

*Your*

An Argus Specialist Publication

October 1983

£1.00

# COMMODORE

**YOUR BEST INDEPENDENT COMMODORE MAGAZINE**

**Build a  
better**

**Going Forth -  
a round up of  
packages for  
your 64**

**Operation  
C-16;  
Complete  
memory map**

**Crowther's  
latest  
blockbuster -  
reviewed**

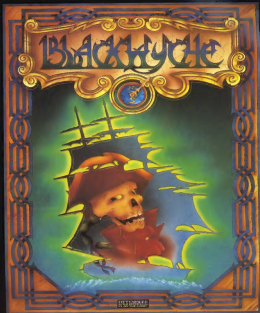


**Competition - 30 copies of  
William Wobbler must be won**

**FIRST  
BIRTHDAY EDITION**



COMMODORE 64



"BLACKTYCHE" recommended retail price £9.95 inc VAT.  
Available from W.H. SMITHS, HOBBS, J. MENZIES, WOOLWORTHS  
and all good software retail outlets. Also available from  
ULTIMATE PLAY THE GAME, The Green, Ashby-de-la-Zouch, Leicestershire LE6 5JU  
(P&P are included) Tel: 0530 411485



# SKYFOX

**NOW  
AVAILABLE  
ON  
CASSETTE  
FOR  
CBM 64**



**1985 AWARD WINNING  
U.S. TOP 10 HIT**



SEVENTH FLEET ATTACK -  
BATTLE OF MIDWAY RECREATED



OUT OF YOUR COCKPIT WINDOW  
AS YOU TAKE OFF TO BATTLE  
OUT OF YOUR COCKPIT WINDOW

## THE ACTION-PACKED COMBAT SIMULATION

"SKYFOX" is the most realistic and inspiring combat simulation you've ever seen on your computer. "SKYFOX" special features include...

- ▶ **AIR TO AIR & AIR TO GROUND COMBAT**
- ▶ **3D SCROLLING FLIGHT SIMULATION**
- ▶ **16 SCENARIOS** (from training mission to massive invasion)
- ▶ **5 SKILL LEVELS** (from cadet to ace of the battle)
- ▶ **AUTO PILOT FEATURE FOR LAUNCH & ENEMY SEARCHES**
- ▶ **ON BOARD & BASE TACTICAL COMBAT COMPUTERS**
- ▶ **CONTINUOUS LASER CANNONS**
- ▶ **HEAT-SEEKING & GUIDED MISSILES**
- ▶ **JOYSTICK & KEYBOARD CONTROLS**

**"SKYFOX" NOW - GET "SKYFOX" NOW - GET "SKYFOX"**

Available for the Apple II, II+, IIx, IIfx, & Commodore 64/128 on Disk, and NOW CBM 64 Cassette

Available from all good software retailers - if it's not there, please order it - and case of difficulty send your credit charged cheque P.O. made out to: Aridola M.U.K. Ltd., including your own name and address to: Aridola M.U.K. Ltd., Suite M2702, Asphodel House, Palace Street, London SW1W 0PS.

▶ **SKYFOX** - Commodore Cassette £9.95, Disk £12.95 ▶ **SKYFOX** - Apple Disk £17.95

**aridola**

BEST PERFORMANCE PROGRAMS







#### VOLUME 2 NUMBER 2

Editor: David Cutler  
Assistant Editor: Mark Lacey  
Software Reviews: John  
O'Connor  
Advertisement Manager: Mike  
Squire  
Advertisement Copy Control:  
Lynn Champion  
Group Editor: Wendy J. Palmer  
Publishing Director: Peter  
Williams  
Organisation, Design, Typetting:  
Design: Mike Dwyer

Editorial & Advertisement Office  
No 1 Golden Square  
London W1R 3JS  
Telephone: 01-477 8626  
Telex: 807086

Your Commodore is a monthly  
requirement appearing on the first  
Friday of each month.

Distribution by Argus Press  
Ltd & Distribution Ltd, 11-13  
Paul Street, London EC4A 3DF  
Printed by: Alden Press  
& Son Ltd, Trill, Newcastle  
upon Tyne

Subscription rates apply  
only to those  
Commodore Subscriptions  
Department, Inland Ltd, News  
House, 175 The Highway,  
Hemel Hempstead, Herts, HP1  
1JL.

The contents of this publication  
including all articles, designs,  
plans, drawings and programs  
and all copyright and other  
intellectual property rights  
therein belong to Argus  
Specialist Publications Limited.  
All rights conferred by the Law  
of Copyright and other  
intellectual property rights and  
by virtue of international  
conventions are  
expressly reserved to Argus  
Specialist Publications Limited  
and any reproduction requires  
the prior written consent of the  
Company. © 1983 Argus  
Specialist Publications Limited.



## FEATURES

### NUTS AND BOLTS OF THE C-16 38

We take a look at the C-16 operating system.

### SPOTS BEFORE YOUR EYES 46

Take on your computer with this game for the  
C16.



### ALL MIXED UP

### ALL MIXED UP 74

Can the best of both worlds in our article on  
machine code and Basic.

### DISK ORDERLY 78

Keep better track of your files with this little  
utility.



## COMPETITION

### COMPETITION

32

Your chance to win a copy of our game of the month.

## SERIES

### TEACHER'S PET

16

Your Commodore is the mafia classroom.

### BUILDING ON BASIC

19

Our new series expanding Basic on your C64.

## REVIEWS

### GOING FORTH

35

Some handy packages to speed up your programming.

### SUPER BOX 64

66

No unplugging with this handy peripheral.

### PRINT IN

76

We take a look at the Macintosh's interface for the Commodore 64.



## REGULARS

### DATA

### STATEMENTS

6

### GAME OF THE MONTH

10

### IN ARCADIA

13

### LISTINGS

14

### SOFTWARE CHART

18

### BUSINESS FILE

29

### COMMUNICATIONS CORNER

30

### BASIC FACTS

43

### INPUT/OUTPUT

54

### SOFTWARE SPOTLIGHT

58

### SENSE OF ADVENTURE

63

### MASTERING MACHINE CODE

69



# CONTENTS



# E-DATA STATEMENTS

Is it a bird? Is it a plane...

The 8th  
**Personal  
Computer  
Show**

**4-8 SEPTEMBER 1985  
OLYMPIA, LONDON**

Sponsored by Personal Computer World

THE PERSONAL COMPUTER WORLD Show is the venue for the preview of Beyond's latest title - *Spectrum*. The official launch date is set for October and the game will be available on the C64 on both cassette and disk.

Beyond's managing director, Bill Delaney, is very confident that the game will be a resounding success. He said: "The game has appeal for everyone - easy to understand with plenty of action and interest for the most avid games fan."

## Earn your stripes

If YOU RATE YOUR PROGRAMMING skills highly then you may be interested in doing some professional work for Tigris Marketing.

The company is often approached by publishers and asked to reprogramme programmes to make Tigris marketing more commercially viable software.

If you specialise in music, adventure, arcade games, strategy, graphics

animation or sports then send details to Tigris Marketing and don't forget to say whether you have any professional experience or not.

Contact: **John Garrett**, Tigris Marketing, Suite 12, Marston Hall, 66 Upper Richmond Rd, Putney, London SW15 2BQ or phone: 01 871 7156/7.

## Quiz kids

THE BCS/COMMODORE SCHOOLS Computer Quiz has been won by Marnick School. The members of the winning team were Greg Michael, Chris Purvis and Tom Duffy. They beat a team from Woodlands High School and received the first prize of £1800-worth of computer equipment for their school.

The final was hosted by Michael Roaf - a long standing TV quizmaster. It took place at the University of Reading and Dr Brian Page, the vice-chancellor, who is also President of the BCS, presented the Momentum Award to Chris Purvis and Desmond Wilson who received the highest scores of the day.

Team members also received individual prizes of a voucher for a free week in Cairn Beaumont on a computer course.

Over 400 schools entered and Commodore's Computers in Schools scheme funded 600 computers to educational establishments for the Quiz. The schools now have the chance to buy the equipment at half price.





## Join the rat race



**HODDER AND STOUGHTON SOFT.** have just now brought an exceptional horse to your computer screen with a new adaptation of James Herriot's best-selling novel, *The Rat*.

Black thorny rats are tearing the streets of London and you are the person unfortunate enough to have been picked to save the world - again. From your control centre you deploy your forces to combat the menace and then you must venture abroad so you can assess the situation first hand. This section of the game leads you through the streets of London, along the banks of canals and into devilish areas.

According to the makers, the game is a combination of strategy and adventure. The adventure section contains animated graphics.

Available now on the Commodore 128, Contact: Hodder and Stoughton Software, PO Box 781, Charnock Green, Sarncliffe, Leeds LS4 2YD.

## Burnt into memory

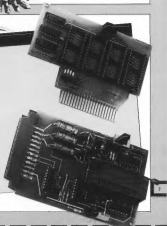
A NEW EPROM BURNER SPECIALLY designed for the C64 and T80, has recently been released by Lightwave Systems. Lightwave also supplies 8K and 128K mother boards for use with the EPROM burner.

EPROM modules enable the user to dispense with ROM chips from cassette or disk and are not harmed by magnetic radiation or ground.

It is possible to erase the programme on the EPROM with ultra-violet radiation and it can then be re-programmed several times.

A software supported register makes the EPROM programs available to the computer at three different address sectors in ROM sections.

Full instructions and operating and control software are supplied with the product. The EPROM burner costs £34.95. The 128K mother board is £24.95 and the 8K board, £12.95. For further information contact: Lightwave Systems, 2 Midway Rd, Liscard, Wirral, Merseyside L44 1AL.





## Fail move!

THE QUICKDISC<sup>®</sup> FROM IVEHAM Micro is a cartridge based enhancement program for the 5.25 disk drive.

Iveham claims many virtues for its new product including fast LOAD and SAVE - four to five times faster - fast FORMAT, fast file save and fast disk backup on unprotected software.

The utility costs £29.95 and is available from Iveham Micro Centre, Bridge St, Iveham, Dorset BH21 4BT.

## Showing off

THERE'S A HUGE VARIETY OF MICRO peripherals on display at the Personal Computer World Show, this year. The show is at London's Olympia and open to the public Friday 4 to Sunday 8 September.

Exhibitors include many big names from the computing world including Amstrad, Atari, Apple and of course Commodore. And there will also be all the latest software on show.

The show occupies two major halls at the exhibition centre and is the biggest so far. All aspects of the computer industry are being covered from home to business, industry to education.

A special attraction is Tomorrow's Africa House which will incorporate all the latest technology in a domestic setting. There's fax, tv, video and computer linked video-disk plus examples of uses for computers in work and leisure.

There's also a special educational section which illustrates the uses of computers in this area.

For those who find themselves spoilt for choice when buying a computer there is a team from the NACC Microcomputers Centre and members will be giving help and advice plus daily seminars on buying a computer for business use.

Major software exhibitors include Arrivision, Microsoft, Martek and Wiley-Cen and in the Commodore stand you'll get another chance to have a look at Commodore's new 128.

The show is open from 10am to 7pm except Sunday when it closes at 5pm. Admission is £2 and tickets are available in advance from PCW Show, 11 Manchester Square, London W1M 1AB.

There will also be an Argus Special Publications stand at the Show which will display Argus' wide range of computing publications covering most aspects of the home micro scene.

Come along and see us and have a browse.



## The Ultimate in skullduggery

Ultimate has a new game for the C64 entitled Blackeye.

The arcade game continues the adventures of Sir Arthur Pennington in an ocean going quest in which he must find

the Crystal Skull of Teels on an island shaped like a skull.

The game will retail at £29.95 and Ultimate's address is The Green, Ashley St, South, Essex SS16 5BA.

## Eyes down



CR1, A FIRM ALREADY WELL KNOWN for its software, has now moved into the peripherals market with the launch of the video Designer Module for the C64.

The unit allows video signals to be displayed on the screen, stored to disk and printed out.

The Austrian designed Designer is compatible with a normal video camera, video surveillance camera, video recorder, video-tapes or a TV or the output from a weather satellite receiver which is video compatible.

The product displays an image on the screen and the user can pan, with the aid

of the cursor keys, to view the whole picture. Initially in four shades of grey, colours can be incorporated using the function keys. Using a lightpen, sections of the picture can be isolated and used later as user defined graphics.

The Designer retails at £149.95 and CR1 claims it brings sophistication and ease of use, previously out of reach of the home computer owner.

Contact: CR1, CR1 Ltd, 1 Kings Yard, Carpenter's Rd, London E15 2JD.



## From bikes to boxing

ACTIVISION HAS PRODUCED A COMPREHENSIVE range of games for the autumn varying from a simulation of the Tour de France to the discovery of a boy being living inside your PC.

The company has signed up world champion boxer, Barry McGuigan, to endorse its new boxing game. Released at the end of August, Activision claims that the game "incorporates his fighting style and unique skills right into the gameplay."

The Great American Cross Country Race is a driving simulation which takes you on a road to coast trip across the USA. Terrain, time of day, weather reports and distance must all be taken into account so there's plenty to think about. You must also watch your fuel and look out for police speed traps.

Somebody's in My Computer is a totally new idea in games. According to Activision, a little known fact is that there is a tiny person living inside every computer. Activision now has the technology to enable you to see and make contact with this little friend. You can tell him to do things, take care of him or just watch as he goes about his business. And so two are the same.

Get on your bike and take part in the Tour de France, the official version of the great French pedalling marathon. The



game relies on joystick dexterity to pedal, steer, brake and change gear around the 16 stage circuit. And if you think all sounds too down to earth for you then one of Activision's other offerings may be your cup of tea. Rescue on Procturus is a space pilot simulation which lands you with the difficult task of saving your fellow pilots who have been stranded on the hostile planet, Procturus.

For utility enthusiasts, Activision is offering the chance to design your own games the easy way with GameMaker. Designed by Gary Kachner, the makers claim that it can help would-be programmers to create almost any game imaginable.

All the new titles are £9.99 on cassette and £14.99 on disk except Game Maker which is £14.99 and £19.99 respectively.

A similar product, recently released, is Computereyes from Stern Computing in London.

Computereyes is a video expansion for the C64 and enables images to be captured on screen and then dumped to a printer.

It connects the video source to the C64 via the Local BUS Port and a grey scale picture appears on the screen in about six seconds.

Software is provided which includes machine language image capture routines, a menu-driven interactive and image packaging routines.

According to Stern Computing, there are numerous uses for the product: pattern recognition, security, spatial measurement and education.

Computereyes costs £109. A demo disk is available for £19 and a source code disk for £49. Stern Computing can be contacted at 218A Queens Avenue, Duxford CB22 1HQ.





William Wobbler is the latest  
Tony Crowther creation from  
Wizard Development. Jayne  
Goin has been delving into its  
secrets for your  
enlightenment.

# GAME OF THE MONTH



THE QUALITY OF C64 GAMES HAS IMPROVED in sophistication over the past year, mainly as a result of the American invasion of the British market. Now our own software houses are fighting back.

William Wobbler by Wizard Development's Tony Crowther illustrates this upward curve by exhibiting not only graphical complexity but also a well-planned gaming system.

The hero of this new action adventure is an odd-looking, slinking character who shuffles across the screen in search of small pieces of a larger puzzle, the solution of which carries a prize worth £1000. The only way to find out what the prize will be is to buy the game and solve the puzzle because the pieces form a picture of the prize.

William's world is a labyrinth of tunnels and passages which are entered by jumping down, hollowed out from walls. There are 10 ranges in all led out across the length of almost as many screens. As William walks his head wobbles up and down and although he can duck under flying frogs and falling rocks his head has a nasty habit of wobbling back up quite quickly so resting is vital.

On entering a hole at the beginning of the game the hole is often blocked by a door which can only be destroyed by the glowing orb hidden in the only droobles hole in the game. The pathway to the orb

is punctuated with small pools of water and stepping stones which William must jump if he is to succeed. Touching the water literally makes him fall to pieces and as if that wasn't enough, there are falling rocks to be ducked.

The only way to enter another hole after negotiating your way back from the orb's hiding place is to jump down another hole to the lowest passageway which is insulated by flying frogs and snakes which mean instant deactivation for William if he's not careful. At the end of this passage is a rope to climb but players will also see a signpost marked "Save Game", following along in the direction of the sign brings William to a disk drive which allows you to save your position in rope but first you must find the disk which operates the drive. Nothing is easy in this game.

On reaching the surface again, William has to get down another very-long without being let in the back of the neck by a pesky bluebird or by being clubbed by the club carried by his more than one enemy.

When William needs any object which he has collected, it appears automatically, demonstrating whether or not the thing you are facing is dangerous or not. Of course this means the object has to be picked up!

This is the major difficulty because you only get one life per game and a final

choice means the loss of all your objects and a restart to the game, unless you have found the disk which does not appear until you have consumed well over the maze. There is also an object in one of the caves which will make you drop everything, you are carrying and replace them where they were originally.

One of the problems to be overcome would make Indiana Jones cringe. A gigantic red ball rolls towards William, totally blocking the tunnel. There is a way to pass this point but I'm not allowed to divulge the secret, however I will say that there is a key object hidden on the other side of the ball.

Some of the tunnels end in designs which are always locked and frequently badly mapped. The map could be on the outside or the inside and the keys are hidden elsewhere in the labyrinth. For example, one key hangs from the roof of a cave. To reach it William must have a piece of rope but no, no! no! no! blocks off the only exit. To escape he must shoot at a target but this implies the ownership of a gun. No gun, no chance.

Scattered liberally through the caves are the clues for the competition and a tally is kept of the number found on a wobbly scrolling scoreboard at the bottom of the screen.

This game makes full use of the 64 and it is renowned that there are only a handful of bytes left unused in the mega-adventure which will become an essential addition to any game player's library.

=====







# 1985 THE YEAR OF COMMUNICATIONS

See your  
Commodore 64  
FREE

A whole new world awaits you and your 64. The world of *Planet*, *Marbles*, *Vivaria*, *Horrors*, *Cinemas*, *Comcast* (Commodore's own database etc.). This is the world of product reviews, multi-theme games, home banking, business information, commodity futures, armchair shopping, real time construction, elections, motion boards, helpful tips, ADPUG, news, News-Link, *Maxima*, *Chuzzle*, *Galaxy* etc. Tunes the world of televisions with programmes to download into your 64 and load to disc or tape. Many third software programmes are provided free of charge or at nominal cost.

Dialog with over 600 exclusive databases on call and instant unlimited information on just about any subject is accessible from your 64, together with 87 Gold, *Boyle*, *Cine* by *Cine* etc., with electronic stock, business services and sales facilities.

You can be in thousands in your area for news, news, club news, orders, plans, plans, helpful tips etc. It's a whole new experience.

Try our first step. Make a special for your 64. Communicate with *Maxima* House and you'll soon be talking to other 64 users, asking for other music and missing files, tips, programmes or just chatting.

*Maxima* House are the largest suppliers of Commodore communications equipment, the best supply complete systems for the Pet 2000, 5000, 6000 and 8000 series.

The most popular 64 microchip in use today, will link you to *Pross*, *Telecom*, *800*, *Maxima*, *276*, *Horrors*, *Foreman*, *Cinemas* & *Comcast* etc. From *Maxima* House the complete system will offer at the incredible price of £99.95 inc. VAT, p & p.

Optional terminal software to link you to 87 Gold, *Boyle*, *Cine*, *Cine* to *Cine*, *Boyle* and *Boyle* boards is available at £29.95 inc. VAT, p & p.

Complete Commodore 64 microchip.

special price **£99.95** inc. VAT p & p  
for a limited period only

**Special, special offer, order your 64 microchip now and get a FREE quarter's subscription to *Maxima*, 800 and *Maxima* and *Boyle* (from Mac accounting).**

Modern House (London) One (Laser) One (DMA) 200, 201, 202, 203, 204

Please send me \_\_\_\_\_ Commodore 64 microchip £99.95 inc. VAT

I enclose £\_\_\_\_\_ Optional terminal software £29.95

I enclose £\_\_\_\_\_ 87 by *Boyle* etc.

Name \_\_\_\_\_

Address \_\_\_\_\_

Telephone No. \_\_\_\_\_

Please allow 30 days delivery

Please fill in by 15th March 1985

YD 10/85

**MODEM  
HOUSE**  
Fourth One, Laser, One (DMA) 200, 201, 202, 203, 204





# WIN·A·R·C·A·D·I·A

I think it's the best, but that's just my opinion. Try it yourself, when it comes out next time.

## Bob's six uncle

I've been playing *Bounty Bob Strikes Back!* (Big Blue/US Gold) lately, and I have achieved a certain competence: that is to say, I've reached the elusive 30,000 score, which, no, my mild-mannered accountant's reasonable score on most games (good rule of thumb, that). My current score is 48,000, and I got to Bob's Playroom as well (Level 4, I think).

I like this game for several reasons. It plays well, the music is something and not in the least boring, and it has the most amazing hi-score table I've ever seen. The best routine to enable the little birds to pick up the letters and place them in the table is an awesome piece of programming, and that's before you even get to the game! A lot of games claim "wonder quality," but *Bounty Bob* has it. I'd happily push my 18 pencils into this one.

## Let's do the time warp again II

I keep mentioning the *Rocky Horror Show*, so it's not hard—this Spectrum (arriveth!) that the Cretaceous version for some time. All my observations on the game had until now been based on this cheaper inferior version. I can report with pleasure that the new CD version is better: it's more difficult to complete the music is what a friend of mine would call less "good," and the sound effects are more than appropriate. But (and as you can see, that's a big but) the game's level 4 down a bit. I've seen multicolored games that would knock yer eyes out, but these are clumsy to say the least, and more multicolored to say the most... Otherwise, a "super game" and certainly well worth a look.

## Hot tips

*Bounty Bob*: The way the joystick controls Bob can be very precise. To jump onto a close platform push the fire button and just nudge the joystick in the desired direction on his way down. Ignore the meters and only kill them if they're getting in your way. Highest scoring strategy: difficulty or medium, bonus at 10,000, and lives lost.

*Transatlantic*: Make the one, two, three when I can give you a bit less all the way in every round before you move on, so make sure I can't multiply again.

*Rock'n'Roll*: Make good use of the Practice Mode (unlocked, level 30, see

my letter to *Before You Begin* a game).

## Better luck next time dept.

*Football* (Activision) promised to be a really good game. The graphics are "bad," but the sound is fairly good. Just it's so "boring" (boring) who cares whether you make it to the top of the screen or not? Activision should really make the game more entertaining, so I really couldn't bear to play this game for more than 10 minutes before I ran and loaded *Bounty Bob* again. Try it out in the shop before you buy it, and see what you think.

*Breakforce* (EPIC/CDI) should be very good, but I couldn't shake the feeling of playing a sort of breakdancing space invader, and paying a lot of money for the privilege! Technically it's very good, program, but largely uninteresting and, I suspect, just a piece of borderline hopping. *Apes* doesn't need to do this. It's a good enough company to make much more original games, without this kind of cheat sheet.

## Ooops! my mistake dept.

My previous apology in English below it. Last month I featured *Henry's House* in my *Transatlantic* Premier issue—I was misinformed as they are actually a bunch of Manchester lads, meaning a trap I for ever allow fall into; just because a piece of software has superb state-of-the-art graphics doesn't mean that the art piece of software comes from across the Atlantic. There is a great deal of world leading programming talent on this island (but that we don't know about yet too) and I promise I won't forget that from now on. (I like 100 lives, *Flippo*—Ed.)

I reserve my apology of last month. *Henry's House* is a very good game. Support your local programmer.

## Mailroom Special Message

Send in your hints, tips, cheats, bugs and hi-score lists to Flippo@epic—Free Call or on. Argus Specialist Publications—No. 1, Golden Square, London W1R 3AB, and use your name in print.

## This month Bounty Bob

comes out on top and

Rock'n'Roll rocks into

Acadia, as Flippo gives you

the low down.

GREETINGS, ACTIONFANS! *WIN·A·R·C·A·D·I·A* took this month with some fabulous tips and tricks for your own arcade games programs, plus a gender at some of the best offerings to hit the shops, and strategies for some new games.

## Poke in the eye dept.

Here are some interesting POKES for you to use in your games programs to discover the chances of any unscrupulous game, or, may be cheating your game and cheating to:

POKE	Function
75,00	Joystick LEFT
71,50	Joystick LEFT
69,00	Joystick SAVE
65,00	Joystick SAVE
61,00	Joystick LOAD
57,00	Joystick LOAD
53,00	Joystick RUN/STOP
49,00	Joystick RUN/STOP
45,00	Joystick RSTOR1
41,00	Joystick RSTOR2
37,00	Changes character colour when using M0, M1 & 7
33,00	removes all the lives M0 in a
29,00	Basic program
25,00	push them all back

This could just as easily put these into an assembly language program as you could in a Basic one.

## Transatlantic Preview

*Rock and Roll* (Konami/Atari) is the best game I've played since *Daddy* (Transpac) first there a year. This is a wonderful mix of the real arcade machine which started this whole sports game work in the first place. You can play against the computer or a real opponent, with a split screen depicting each player's progress, a lot like *Pong* II.





Listings will be much easier to  
 enter with our new system.

COMMODORE LISTINGS ARE RATHER well known for the horrible little black boxes that always abound. Unfortunately the graphics characters which are used to represent graphic and control characters do not reproduce very well and they are also difficult to find on the Commodore keyboard.

For the reason that Commodore started to precede any control characters with a \$ (dollar) symbol rather than a tilde (~) which explained exactly what the black boxes were meant to be. Unfortunately the graphics characters were not documented and these still cause some confusion. For this reason we are starting to use a new method for marking the control and graphics characters in our listings.

In future all control and graphics commands will be replaced by references within square brackets. This new style is not typed out as printed in the magazine but rather the corresponding key or keys on the keyboard are pressed. For example [RIGHT] means press the cursor right key, you do not type in [RIGHT]. All of the keywords, what keys to press and how they are shown on the screen are shown below.

# LISTINGS

Any character that is accessed by pressing shift and letter will be printed as (x) [LTTTRR]

(x) A)  
(x) C)

shift and A  
shift & C

Any character that is accessed by pressing the Commodore key and a letter will be passed as (x) [LTTTRR]

(x) A)  
(x) C)

Commodore & A  
Commodore & C

(x) 1)

Any control key will be printed out as a number. For example [001]. Control codes are accessed by pressing the CTRL key and a letter at the same time [001] is CTRL & A, [002] is CTRL & B etc. See the manual for more information about control codes.

[001]  
[002]

CTRL & A  
CTRL & B

Reference	Symbol	what to press
[RIGHT]		left/right
[LEFT]		shift left/right
[UP]		Shift & up/down
[DOWN]		up/down
[H]		h
[F]		shift & f
[F1]		f1
[F4]		shift & f4

Reference	Symbol	what to press
[TS]		tab
[PA]		shift & TS
[RT]		rt
[R]		shift & TS
[CLEAR]		shift & CLR (HOME)
[HOME]		CLR/HOME
[F5]		CTRL & F
[F6]		CTRL & F

Reference	Symbol	what to press
[BLACK]		CTRL & 1
[WHITE]		CTRL & 2
[RED]		CTRL & 3
[CYAN]		CTRL & 4
[BLUE]		CTRL & 5
[GREEN]		CTRL & 6
[BELL]		CTRL & 7
[YELLOW]		CTRL & 8





Mr. Software



# FREE GAMES!



YES... place an order for our Commodore 64 Magazine and we will send you one of the above games absolutely free! We will also enroll you in our Discount Software Club, enabling you to purchase software at huge savings over shop prices! We offer all the latest titles at substantial discounts such as: Frank Bruno's Boxing at £3.95 (saving £3.00), Frankie Goes To Hollywood at £8.95 (saving another £3.00) or Spy Versus Spy at £8.95 (saving yet another £3.00)! We also have a number of older titles at even bigger discounts, such as: Attack of the Mutant Camels at £2.95, Chinese Juggler at £1.95, Scuba Dive at £2.95 etc! In all we have over 200 titles available for your computer.



## SIX MAGAZINES A YEAR

A subscription of just £5.00 will bring you six magazines a year, (through the post). Each magazine is packed with reviews of all the latest software, reader's games playing tips, competitions to enter, our reviewer's personal choices and a summary of how the other magazines rate the new releases. **HARDWARE AND UTILITIES SUPPLEMENT**

Every new subscriber to the magazine gets a 32-page review of some of the more popular utilities (such as word processing software) and hardware including joysticks.

## HOTLINE KEEPS YOU UP TO DATE

Subscribers to the magazine can phone our 24 hour Hotline for up to the minute news of the latest releases! A three minute recording, updated weekly, details the new titles and their prices. Club members may order by phone too!

## EASY ORDERING

You can order from our substantial range as and when you like. There are no minimum order requirements (if you don't have to buy anything at all). When you do order we do our best to dispatch your goods within a week... often quicker!

## DOIT TAKE OUR WORD?

Don't miss some of our subscribers say: "Many congratulations on your excellent reviews & proper quality!"  
Mark A. P. Colclough, Toronto, Canada

"I have seen Club in operation, and... you are about after each magazine!"  
David Davies, Manchester

Many thanks to you for...

Mr. Software

ELVIN HOUSE, POTTERIDGE  
AVENUE, HIGH WYCOMBE,  
HP12 3EG. 0494 450617



Mr. Software

## SUBSCRIPTION FORM TO NEWS

with to receive a subscription to the Mr. Software

(or Mr. Software)

for the next 12 months for which payment of £5.00 (including postage and the first magazine) should be sent to your nearest Mr. Software Club agent or to a retail outlet where you can find Mr. Software

Magazine

to

Name (PLEASE PRINT)

Address

Postcode

Customer Period Order for

Charge the above to my account

Mr. Software is a 12-page color magazine. Other titles are 11 pages. £2.95 a month (inc. 1st issue). For the 1st issue







# STEP INTO THE FUTURE WITH "C POWER" —

the C Language Compiler for the COMMODORE 64

**ATTENTION ALL  
COMMODORE 64  
PROGRAMMERS**

Your personal guide to "C"  
The world's most effective and  
versatile programming language  
C POWER is a full  
implementation, standard C  
language compiler for the  
COMMODORE 64 - plus 100-  
+ full 500 pages!

This powerful yet easy-to-use  
compiler, with:

- A diskette and a disk
- A 501 page C language  
Tutorial, "C Power Plus" - a  
friendly, step-by-step guide to  
writing and using C -
- A 64 page user's guide  
complete with listing and  
explanation

## C POWER GIVES YOU MORE:

- Front-end editor, Source  
checking editor
- Shell, Environment Manager
- assembler Translator
- Standard System and Math  
Language Routines
- List of 100 key routines
- Sample programs
- automatic, low-level compiling  
and cut-and-paste facilities
- Value for money at £119.95

## C POWER IS:

- Ready-to-go—everything you  
will use in C Power belongs to  
you.
- On-line: the C Power system  
works even operates 7 days  
a week, 24 hours a day - to  
update your function libraries  
and answer your questions
- A fully functioning,  
integrated, complete C  
language Compiler for the  
COMMODORE 64

## THE EXPERTS AGREE:

"an excellent package  
deserving to be in everyone's  
software library"

John Clark, *ANAL*, 12  
July 1983

"a refreshing change... with the  
speed of machine language and  
the programmability of BASIC"

John Clark, *ANAL*, 14  
July 1983



## ORDER NOW

For your C POWER (R) PACKAGE send this form to:

**Prison Publishing Ltd**  
138 Long Acree  
London  
WC1E 8AN

Please send me C POWER, price £119.95 (inc VAT)  
39% GST 50483 5

I enclose a cheque/credit order (Payable to Prison  
Publishing Ltd £\_\_\_\_\_)

Alternatively, please debit my ☐ ☐ ☐ ☐ ☐ ☐

Account Number: \_\_\_\_\_

Signed: \_\_\_\_\_

Name (as this appears): \_\_\_\_\_

Address: \_\_\_\_\_



# TOP 20 Gallup Software

## COMMODORE 64

Title	Publisher
1 The Way of the Exploding Fist	Atari
2 Elite	Atari
3 Soft Add	Various
4 Pitstop II	US Gold
5 Dambusters	US Gold
6 A View to a Kill	Domark
7 Saboteur	Mastertronic
8 International Basketball	Commodore
9 International Tennis	Commodore
10 Dropzone	US Gold
11 Shadowgate	Beyond
12 Impossible Mission	US Gold
13 Cauldron	Falco
14 Entombed	Ultimate
15 Graham Coxon's Test Crater	Atari
16 Action Biker	Mastertronic
17 Executive Europe	PSI
18 Rocky Horror Show	CBI
19 Jet Set Willy II	Software Projects
20 Basic Lee	US Gold

Retail sales for week ending July 29, 1985.



## VIC 20 Top Ten

Title	Publisher
1 Ring Fit	Mastertronic
2 BP	Mastertronic
3 Rockstar	Mastertronic
4 Football Manager	Addictive
5 Mickey the Brick	Entand
6 Vegas Jackpot	Mastertronic
7 Bullseye	Mastertronic
8 Snakebite	Entand
9 Snakebite	Entand
10 Psycho Shopper	Mastertronic

Retail sales for week ending July 13, 1985.

Compiled by Gallup for the industry's leading magazine, Computer and Software Publishing. For details contact John Ross, Computer and Software Publishing, 222 Regent Street, London W1B 1AB. 01-4754 2911.



In the first of a new series,  
 Nick Hampshire brings you  
 the first pieces of a bigger,  
 better Basic.

# BUILDING ON BASIC



HOW MANY TIMES HAVE YOU thought how nice it would be to have a Basic command to perform a particular function? The kind which would normally require a convoluted piece of Basic code, or a short machine code routine called by a GOTO command. Adding extra commands to Basic on the C64 is not difficult and can make writing files or programs a lot quicker and easier.

There are of course a considerable number of commercially produced extended Basic packages. But, being able to add and create your own commands is much more fun and allows the commands to be tailored to your particular needs.

This article is the first of a series which will include the code for adding a whole range of very useful commands to your Basic interpreter. This article includes the code for the all-important control wedges which allow the new Basic commands to work. It is essential that these routines are in memory before any attempt is made to add commands and routines.

The following routines are the start of the Basic extension code. These are the main control routines which patch the old commands into the C64's Basic. They should be listed in the order they appear in the article and the accompanying listings.

## Initialisation

The section of code contains the initialisation routines and the table of added commands and their vectors. The commands are initialised by calling the add start (MPC2 - 64708) which calculates a random power up. The routines cannot be used with a cartridge in place as they set up the same locations and are not good for the purpose of being placed on a cartridge ROM if required.

The routine labelled 'COLOR' is the power-up routine and the routine called 'WASH' is the RAM routine. The WASH routine makes sure that the function key and basic wedge routines are not

disabled. Note that the table of added commands and their addresses can be changed and added to by the user. The commands given in the table refer to commands which will be added to this package later in the series. Any reader assembling the portion of this code will get a 'Label not present' error from the LADOR table unless a dummy routine is set up for every added Basic command which has not yet been implemented. This routine simply takes the following form.

```
DECODE JWP 14000
```

where DECODE is the label. Thus, since the first two commands are included at the end of this article a total of 18 dummy routines are required at this stage, which can be removed as the appropriate commands are added. Such a set of dummy routines is shown in the file, L8 DUMMY at the end of the listings.

## Crunch to Tokens

This routine is wedged into the Crunch token link at locations \$0264-\$0269 (773-778). Crunch to tokens will take the input line and convert all command words to one (normal) or two (standard) byte token values. This does exactly the same as the original Basic version except that the extended keyword table is checked before the normal Basic table.

Crunch to tokens is done directly after the warm start routine to conserve a carriage return. It makes no difference whether the command is direct mode or for entering, or defining a line in memory.

## Tokens to Text

This routine is wedged into the Parse token link at locations \$026A-\$026F (779-77E). Tokens to text is used in the for command and only to convert any token value greater than 127 for Basic or previously \$42 = 255 for Extended Basic back into the original word and print it to the output device.

## Execute Statement

This routine is wedged into the Start new Basic code link at locations \$026B-\$026E (775-778). This is the control part of the main Basic Interpreter loop and takes a token value and extracts the routine via the vector table in the initialisation.

There is a special case routine for PRINT which uses the same token as its normal Basic but the routine has been rewritten to allow the CTL command.

## Execute Arithmetic

This routine is wedged into the Arithmetic link at locations \$026A-\$026B (775-776).





77h. This routine is used by the user expressions and numbers entered to one of the four arithmetic routines as coded in this passage if the extended basic command is not one of the four arithmetic routines, unless error is output.

## Function keys

This routine is entered into the keyboard setup entry at locations 9000-9009 415 416. The routine checks if the computer is in direct or program mode. If in direct, the normal routine is executed. If in program mode, the cursor flag is checked and if set, the normal routine is entered.

The commands passed as below for one of the four function keys and the shift key. If it is a function key, the test for that key is used (in fact, the line 9004) and put into the keyboard buffer until it equals character 0. A zero here means cursor is moved if not a function key, the normal routine is entered.

## Program User

It is noted we are working into the 1000, I enter at locations 9104-9125 804-815. In direct simulators, the normal input routine find the input device is checked for keyboard. If not found, the normal routine is entered. Direct mode is then checked and if it is found, the normal routine is again entered.

The next part of the routine is copied directly from the normal routine except that the cursor down key is checked not and if found, then the cursor position is checked. If the cursor is up on the bottom line of the screen, the cursor down character is printed. If the cursor is on the bottom line, instead of printing cursor down, the next line number is found and that line used to output device.

NOTE: there is no check for spaces as if you repress a key on the bottom line on the screen, the line is to be wiped out and a line listed if you press the cursor down key, just leave sufficient spaces.

When the last line of the program is listed, the cursor will remain at the end of the line. Cursor down again will produce the message:

```
*****END OF PROGRAM*****
```

after this, the program will start listing from the beginning again.

## Print

The purpose of this routine is to PRINT characters to the open CATH output channel, usually with three - spaces. The version of PRINT does exactly the

## Basic Loader

```
1000 REM *****
1010 REM #
1020 REM # BASIC LOADER FOR EXTENDED
1030 REM # BASIC COMMAND
1040 REM # PACKAGE
1050 REM #
1060 REM # COPYRIGHT 1985
1070 REM # NICK HAMPSHIRE
1080 REM #
1090 REM *****
1100 PRINT"*****      ENTERING EXTENDED BASIC"
1110 I=0:Z=0:Tab
1120 READ I:IF I=1 THEN I=150
1130 POKE I,A: T=T+A
1140 I=I+1:GOTO 1120
1150 IF I>20000 THEN PRINT"NO CHECKSUM
ERROR: "I"SHOULD BE 20000"
1160 IF I>34510 THEN PRINT"NUMBER OF
VALUES ERROR: "I"SHOULD BE 34510":END
1170 PRINT"NO VALUES ENTERED CORRECTLY"
1180 PRINT"NOT A RUN PRESS ANY KEY"
1190 GETA: IF A<0 THEN I=200:GOTO 1120
1200 SYS 164738
2000 DATA 128,129,57,129,195,194,205
2010 DATA 56,40,129,227,131,164,201
2020 DATA 129,158,130,247,139,59,131
2030 DATA 76,72,178,0,49,234,68
2040 DATA 129,71,204,74,243,145,242
2050 DATA 14,242,88,242,51,243,241
2060 DATA 131,202,241,237,246,62,241
2070 DATA 47,243,68,128,165,244,237
2080 DATA 245,32,189,246,32,225,235
2090 DATA 248,3,76,114,204,32,163
2100 DATA 253,32,34,229,32,93,128
2110 DATA 32,204,255,169,0,133,19
2120 DATA 32,122,166,89,162,128,76
2130 DATA 136,227,162,21,168,128,134
2140 DATA 195,132,186,168,35,177,195
2150 DATA 153,16,3,136,16,248,169
2160 DATA 118,168,131,141,143,2,148
2170 DATA 44,2,56,142,22,208,32
2180 DATA 163,253,32,88,253,32,91
2190 DATA 255,32,93,128,88,32,229
2200 DATA 128,32,191,227,169,128,130
2210 DATA 52,133,54,133,56,169,0
2220 DATA 133,51,133,53,133,55,169
2230 DATA 172,168,128,32,45,228,162
2240 DATA 251,154,208,172,147,13,32
2250 DATA 32,32,32,42,42,42,42,42
2260 DATA 32,68,88,94,69,78,68
2270 DATA 69,68,32,54,52,32,66
2280 DATA 65,83,73,67,32,86,48
2290 DATA 49,32,42,42,42,42,13
2300 DATA 13,32,54,52,75,32,62
```



## Basic Loader

```

2310 DATA65.77.32.83.83.83.84
2320 DATA69.77.32.32.8.162.11
2330 DATA189.9.128.157.8.3.282
2340 DATA16.247.96.82.85.286.67
2350 DATA84.284.65.88.88.88.78
2360 DATA136.65.85.84.287.67.85
2370 DATA84.65.76.79.139.67.72
2380 DATA65.78.71.157.67.72.65
2390 DATA73.286.67.82.85.78.67
2400 DATA288.68.69.76.69.84.197
2410 DATA68.73.83.283.68.79.75
2420 DATA157.68.85.77.288.69.88
2430 DATA69.195.78.73.78.196.71
2440 DATA69.212.75.69.217.77.65
2450 DATA12.77.69.82.71.197.79
2460 DATA76.196.88.79.288.88.85
2470 DATA12.82.69.78.85.77.66
2480 DATA69.218.82.69.88.69.65
2490 DATA12.83.79.82.212.84.82
2500 DATA65.67.69.79.286.84.82
2510 DATA65.67.69.79.78.198.84
2520 DATA89.88.157.85.78.84.73
2530 DATA84.68.69.69.283.72.73
2540 DATA77.69.285.76.79.77.69
2550 DATA285.86.65.82.88.84.218
2560 DATA8.116.134.138.133.119.134
2570 DATA122.134.125.134.128.134.131
2580 DATA134.134.134.137.134.148.134
2590 DATA143.134.146.134.149.134.152
2600 DATA134.155.134.158.134.161.134
2610 DATA164.134.167.134.178.134.173
2620 DATA134.176.134.179.134.182.134
2630 DATA185.134.188.134.191.134.134
2640 DATA134.197.134.288.134.283.134
2650 DATA86.134.166.122.168.4.132
2660 DATA15.189.8.2.16.7.281
2670 DATA55.248.43.232.288.244.281
2680 DATA32.248.36.133.8.281.34
2690 DATA248.71.36.15.112.26.281
2700 DATA63.288.4.169.153.288.18
2710 DATA281.48.144.4.281.68.144
2720 DATA18.76.78.138.163.238.44
2730 DATA5.11.164.113.232.288.153
2740 DATA251.1.281.238.248.49.185
2750 DATA251.1.248.34.56.233.58
2760 DATA248.4.281.73.288.2.133
2770 DATA15.56.233.85.288.174.133
2780 DATA8.189.8.2.248.215.197
2790 DATA8.248.215.288.193.251.1
2800 DATA232.288.248.153.253.1.198
2810 DATA123.169.253.133.122.96.165
2820 DATA11.248.153.251.1.76.287
2830 DATA129.132.113.168.255.134.122
2840 DATA282.169.1.133.11.288.232
2850 DATA189.8.2.56.249.241.128

```

was at the Basic PRINT except that a check has been made for the CTL command to be included. The syntax for this modified command is exactly the same as that of the basic PRINT command.

## The First Extended Basic Command

## CTL

Abbreviated entry: C@h@T

Altered Basic abbreviation: none

Notes: Hex \$1140 (Decimal 1384)

Modes: Direct, program, and in PR#T systems

Purpose: To replace cursor and colour characters, screen and border colours, improving the ability to position the cursor anywhere on the screen. If the value is not specified, the current value is used.

Syntax: CTL (X)(Y)(C)(B)(L)@

Where x is the column position of the cursor (0-24), y is the row position of the cursor (0-24), c is the cursor colour, b is the screen colour, l is the border colour (0-15), and @ is a flag for clearing the screen (0 - no, 1 - yes).

Error Syntax error - if the syntax is not as above.

Illegal quantity - if the values are out of range.

Use: CTL is a powerful screen handling routine. Cursor, screen, and border colour can be set with a number (0-15), and the cursor can be positioned anywhere on the screen by entering the x, y, c, b, l, @. There is also a screen clear flag (l), if set to '1', will clear the screen before positioning the cursor. To make it easier to describe, here are a few examples and what they do.

CTL 128 positions cursor at middle of current line

CTL 800 moves cursor to 800 (screen position)

CTL 1 sets cursor colour to white

CTL 1 sets screen colour to black

CTL 15 sets border colour to black

CTL 1 clears screen leaving cursor at current position

CTL (20)(20)(15) clears screen (15 sets screen to black 15, border to medium grey (11), cursor colour to green (5), and cursor position to column 20, row 12).

To print something at a specified location on the screen,

PRINT CTL (x,y) "text" CTL (x,y) "more text"

Routine entry point: \$1848

Routine operation: The current settings of the first parameters are read and the screen clear flag is set to 0. The open brackets character is inserted just and each of the six values is read if present, checking to see if there is a closing bracket. When the closing bracket is found, the screen is cleared if the flag is set to 1, and the other values are stored to their own locations.



# Basic Leader

2860	DATA240.245.201.129.240.156.146	3420	DATA169.55.193.1.165.203.133
2870	DATA122.239.111.200.165.240.128	3430	DATA197.173.141.2.141.242.2
2880	DATA16.250.185.241.129.208.228	3440	DATA96.165.173.209.4.165.157
2890	DATA160.0.132.11.136.166.122	3450	DATA208.2.76.87.241.165.211
2900	DATA202.208.232.189.0.2.55	3460	DATA133.202.165.214.133.201.157
2910	DATA249.158.160.240.245.201.129	3470	DATA72.129.72.165.209.240.6
2920	DATA208.3.76.255.129.166.122	3480	DATA76.59.209.32.22.231.165
2930	DATA238.11.200.165.157.160.16	3490	DATA138.133.204.141.146.2.240
2940	DATA250.185.158.168.208.235.189	3500	DATA247.128.165.207.240.12.165
2950	DATA0.2.76.1.130.48.3	3510	DATA206.174.135.2.160.0.132
2960	DATA6.243.166.201.255.240.249	3520	DATA207.32.19.234.32.180.229
2970	DATA36.15.48.245.201.235.240	3530	DATA201.131.208.16.162.9.120
2980	DATA5.32.217.130.48.3.32	3540	DATA134.198.189.230.236.157.112
2990	DATA196.130.76.239.166.200.177	3550	DATA2.202.208.247.240.207.201
3000	DATA95.170.132.73.169.275.202	3560	DATA13.209.3.76.2.230.201
3010	DATA240.8.200.195.241.129.16	3570	DATA17.200.193.166.214.224.24
3020	DATA250.48.245.200.185.241.128	3580	DATA40.3.76.15.132.163.24
3030	DATA48.5.32.218.255.208.245	3590	DATA160.0.24.32.240.255.230
3040	DATA96.56.233.127.170.132.73	3600	DATA0.209.2.230.21.37.19
3050	DATA168.255.202.240.8.200.185	3610	DATA162.160.1.157.95.203.16
3060	DATA150.160.16.250.48.245.200	3620	DATA169.255.133.20.133.21.163
3070	DATA185.150.160.48.230.12.210	3630	DATA185.169.132.32.18.171.74
3080	DATA255.238.245.32.115.0.241	3640	DATA18.132.160.2.17.95.133
3090	DATA230.240.10.201.153.240.18	3650	DATA90.208.177.95.123.21.162
3100	DATA12.121.0.76.231.167.32	3660	DATA162.143.0.3.163.132.141
3110	DATA14.131.76.174.167.230.122	3670	DATA1.3.104.141.103.132.104
3120	DATA200.2.230.123.168.0.177	3680	DATA141.184.132.160.1.133.15
3130	DATA122.56.233.1.10.168.185	3690	DATA76.215.166.163.133.141.0
3140	DATA138.129.72.185.137.129.72	3700	DATA3.169.227.141.1.3.173
3150	DATA76.115.0.32.46.131.76	3710	DATA184.132.72.173.163.132.12
3160	DATA174.167.173.58.131.72.173	3720	DATA76.18.132.0.8.11.13
3170	DATA57.131.72.76.115.0.235	3730	DATA18.42.42.42.42.42.42
3180	DATA132.169.8.133.13.32.115	3740	DATA42.42.42.42.42.42.42
3190	DATA8.201.238.248.6.32.121	3750	DATA0.78.0.32.79.79.32
3200	DATA8.76.141.174.238.122.208	3760	DATA0.32.79.71.82.65.77
3210	DATA2.208.123.160.0.177.122	3770	DATA13.42.42.42.42.42.42
3220	DATA201.29.176.3.76.8.175	3780	DATA42.42.42.42.42.42.13
3230	DATA133.36.169.173.72.169.140	3790	DATA0.32.171.32.121.0
3240	DATA72.195.36.165.36.18.178	3800	DATA240.10.240.40.201.163.240
3250	DATA169.138.129.72.183.173.129	3810	DATA207.201.140.24.240.192.141
3260	DATA72.76.115.0.165.157.240	3820	DATA20.208.20.163.1.177.122
3270	DATA16.163.1.36.212.208.10	3830	DATA201.2.200.12.32.115.0
3280	DATA165.203.201.3.144.4.201	3840	DATA12.115.0.32.147.157.76
3290	DATA7.144.3.76.72.235.197	3850	DATA232.132.2.123.0.141.41
3300	DATA197.248.249.169.8.133.252	3860	DATA240.75.201.79.440.97.32
3310	DATA133.251.169.1.44.141.2	3870	DATA159.173.16.13.45.195.1
3320	DATA240.4.169.32.133.251.169	3880	DATA21.184.32.125.150.32.73
3330	DATA191.133.252.163.132.34.181	3890	DATA171.32.59.178.208.164.169
3340	DATA251.133.251.165.203.201.3	3900	DATA1.157.0.2.162.175.180
3350	DATA208.4.163.24.209.18.201	3910	DATA1.163.1.42.0.16.169.13
3360	DATA6.208.4.169.16.208.18	3920	DATA1.71.171.36.19.16.5
3370	DATA201.5.280.4.169.6.203	3930	DATA169.10.32.71.171.71.75
3380	DATA2.169.0.24.101.251.133	3940	DATA26.58.2.240.45.152.54
3390	DATA21.169.0.163.54.133.1	3950	DATA33.10.176.23.23.275.197
3400	DATA177.251.240.8.173.119.2	3960	DATA1.206.15.8.58.23.240
3410	DATA200.192.8.208.244.132.190		







Machine Data Listing

[illegible]



**Abstract**

[illegible]



[illegible][illegible]



### Machine Code Listing

```

000000 000 000000      .GET NEXT CHAP
000001 000 000000      .SET COUNTER TO
        . ADDRESS OF LINE
000002 000 000000
000003 000 000000      . AND NEXT LINE TO
        . LINE
000004 000 000000
000005 000 000000      .GET ADDRESS
000006 000 000000      . ADDRESS OF PROGRAM
        . NO
000007 000 000000      . NEXT LINE NUMBER+
        . NO
000008 000 000000      . CALL USER PART TWO
        . END OF PROGRAM AND
        . GETS RETURNED
000009 000 000000      . SET NEXT CHAP
        . SET CTR NUMBER
000010 000 000000      . NO GET
        . NO GET
000011 000 000000
000012 000 000000      . NO GET
000013 000 000000
000014 000 000000      . RETURN TO LISTED
        . AFTER LIST
000015 000 000000
000016 000 000000      . WHEN I STOP IN
        . THIS LOCATION
000017 000 000000
000018 000 000000
000019 000 000000
000020 000 000000
000021 000 000000
000022 000 000000
000023 000 000000
000024 000 000000
000025 000 000000
000026 000 000000
000027 000 000000
000028 000 000000
000029 000 000000
000030 000 000000
000031 000 000000
000032 000 000000
000033 000 000000
000034 000 000000
000035 000 000000
000036 000 000000
000037 000 000000
000038 000 000000
000039 000 000000
000040 000 000000
000041 000 000000
000042 000 000000
000043 000 000000
000044 000 000000
000045 000 000000
000046 000 000000
000047 000 000000
000048 000 000000
000049 000 000000
000050 000 000000
000051 000 000000
000052 000 000000
000053 000 000000
000054 000 000000
000055 000 000000
000056 000 000000
000057 000 000000
000058 000 000000
000059 000 000000
000060 000 000000
000061 000 000000
000062 000 000000
000063 000 000000
000064 000 000000
000065 000 000000
000066 000 000000
000067 000 000000
000068 000 000000
000069 000 000000
000070 000 000000
000071 000 000000
000072 000 000000
000073 000 000000
000074 000 000000
000075 000 000000
000076 000 000000
000077 000 000000
000078 000 000000
000079 000 000000
000080 000 000000
000081 000 000000
000082 000 000000
000083 000 000000
000084 000 000000
000085 000 000000
000086 000 000000
000087 000 000000
000088 000 000000
000089 000 000000
000090 000 000000
000091 000 000000
000092 000 000000
000093 000 000000
000094 000 000000
000095 000 000000
000096 000 000000
000097 000 000000
000098 000 000000
000099 000 000000

```



## Machine Code Listing

001	LDI R16,0	000	LDI R0,0	0000	LDI R0,0	0000	LDI R0,0
002	LDI R17,0	001	LDI R1,0	0001	LDI R1,0	0001	LDI R1,0
003	LDI R18,0	002	LDI R2,0	0002	LDI R2,0	0002	LDI R2,0
004	LDI R19,0	003	LDI R3,0	0003	LDI R3,0	0003	LDI R3,0
005	LDI R20,0	004	LDI R4,0	0004	LDI R4,0	0004	LDI R4,0
006	LDI R21,0	005	LDI R5,0	0005	LDI R5,0	0005	LDI R5,0
007	LDI R22,0	006	LDI R6,0	0006	LDI R6,0	0006	LDI R6,0
008	LDI R23,0	007	LDI R7,0	0007	LDI R7,0	0007	LDI R7,0
009	LDI R24,0	008	LDI R8,0	0008	LDI R8,0	0008	LDI R8,0
010	LDI R25,0	009	LDI R9,0	0009	LDI R9,0	0009	LDI R9,0
011	LDI R26,0	010	LDI R10,0	0010	LDI R10,0	0010	LDI R10,0
012	LDI R27,0	011	LDI R11,0	0011	LDI R11,0	0011	LDI R11,0
013	LDI R28,0	012	LDI R12,0	0012	LDI R12,0	0012	LDI R12,0
014	LDI R29,0	013	LDI R13,0	0013	LDI R13,0	0013	LDI R13,0
015	LDI R30,0	014	LDI R14,0	0014	LDI R14,0	0014	LDI R14,0
016	LDI R31,0	015	LDI R15,0	0015	LDI R15,0	0015	LDI R15,0
017	LDI R32,0	016	LDI R16,0	0016	LDI R16,0	0016	LDI R16,0
018	LDI R33,0	017	LDI R17,0	0017	LDI R17,0	0017	LDI R17,0
019	LDI R34,0	018	LDI R18,0	0018	LDI R18,0	0018	LDI R18,0
020	LDI R35,0	019	LDI R19,0	0019	LDI R19,0	0019	LDI R19,0
021	LDI R36,0	020	LDI R20,0	0020	LDI R20,0	0020	LDI R20,0
022	LDI R37,0	021	LDI R21,0	0021	LDI R21,0	0021	LDI R21,0
023	LDI R38,0	022	LDI R22,0	0022	LDI R22,0	0022	LDI R22,0
024	LDI R39,0	023	LDI R23,0	0023	LDI R23,0	0023	LDI R23,0
025	LDI R40,0	024	LDI R24,0	0024	LDI R24,0	0024	LDI R24,0
026	LDI R41,0	025	LDI R25,0	0025	LDI R25,0	0025	LDI R25,0
027	LDI R42,0	026	LDI R26,0	0026	LDI R26,0	0026	LDI R26,0
028	LDI R43,0	027	LDI R27,0	0027	LDI R27,0	0027	LDI R27,0
029	LDI R44,0	028	LDI R28,0	0028	LDI R28,0	0028	LDI R28,0
030	LDI R45,0	029	LDI R29,0	0029	LDI R29,0	0029	LDI R29,0
031	LDI R46,0	030	LDI R30,0	0030	LDI R30,0	0030	LDI R30,0
032	LDI R47,0	031	LDI R31,0	0031	LDI R31,0	0031	LDI R31,0
033	LDI R48,0	032	LDI R32,0	0032	LDI R32,0	0032	LDI R32,0
034	LDI R49,0	033	LDI R33,0	0033	LDI R33,0	0033	LDI R33,0
035	LDI R50,0	034	LDI R34,0	0034	LDI R34,0	0034	LDI R34,0
036	LDI R51,0	035	LDI R35,0	0035	LDI R35,0	0035	LDI R35,0
037	LDI R52,0	036	LDI R36,0	0036	LDI R36,0	0036	LDI R36,0
038	LDI R53,0	037	LDI R37,0	0037	LDI R37,0	0037	LDI R37,0
039	LDI R54,0	038	LDI R38,0	0038	LDI R38,0	0038	LDI R38,0
040	LDI R55,0	039	LDI R39,0	0039	LDI R39,0	0039	LDI R39,0
041	LDI R56,0	040	LDI R40,0	0040	LDI R40,0	0040	LDI R40,0
042	LDI R57,0	041	LDI R41,0	0041	LDI R41,0	0041	LDI R41,0
043	LDI R58,0	042	LDI R42,0	0042	LDI R42,0	0042	LDI R42,0
044	LDI R59,0	043	LDI R43,0	0043	LDI R43,0	0043	LDI R43,0
045	LDI R60,0	044	LDI R44,0	0044	LDI R44,0	0044	LDI R44,0
046	LDI R61,0	045	LDI R45,0	0045	LDI R45,0	0045	LDI R45,0
047	LDI R62,0	046	LDI R46,0	0046	LDI R46,0	0046	LDI R46,0
048	LDI R63,0	047	LDI R47,0	0047	LDI R47,0	0047	LDI R47,0
049	LDI R64,0	048	LDI R48,0	0048	LDI R48,0	0048	LDI R48,0
050	LDI R65,0	049	LDI R49,0	0049	LDI R49,0	0049	LDI R49,0
051	LDI R66,0	050	LDI R50,0	0050	LDI R50,0	0050	LDI R50,0
052	LDI R67,0	051	LDI R51,0	0051	LDI R51,0	0051	LDI R51,0
053	LDI R68,0	052	LDI R52,0	0052	LDI R52,0	0052	LDI R52,0
054	LDI R69,0	053	LDI R53,0	0053	LDI R53,0	0053	LDI R53,0
055	LDI R70,0	054	LDI R54,0	0054	LDI R54,0	0054	LDI R54,0
056	LDI R71,0	055	LDI R55,0	0055	LDI R55,0	0055	LDI R55,0
057	LDI R72,0	056	LDI R56,0	0056	LDI R56,0	0056	LDI R56,0
058	LDI R73,0	057	LDI R57,0	0057	LDI R57,0	0057	LDI R57,0
059	LDI R74,0	058	LDI R58,0	0058	LDI R58,0	0058	LDI R58,0
060	LDI R75,0	059	LDI R59,0	0059	LDI R59,0	0059	LDI R59,0
061	LDI R76,0	060	LDI R60,0	0060	LDI R60,0	0060	LDI R60,0
062	LDI R77,0	061	LDI R61,0	0061	LDI R61,0	0061	LDI R61,0
063	LDI R78,0	062	LDI R62,0	0062	LDI R62,0	0062	LDI R62,0
064	LDI R79,0	063	LDI R63,0	0063	LDI R63,0	0063	LDI R63,0
065	LDI R80,0	064	LDI R64,0	0064	LDI R64,0	0064	LDI R64,0
066	LDI R81,0	065	LDI R65,0	0065	LDI R65,0	0065	LDI R65,0
067	LDI R82,0	066	LDI R66,0	0066	LDI R66,0	0066	LDI R66,0
068	LDI R83,0	067	LDI R67,0	0067	LDI R67,0	0067	LDI R67,0
069	LDI R84,0	068	LDI R68,0	0068	LDI R68,0	0068	LDI R68,0
070	LDI R85,0	069	LDI R69,0	0069	LDI R69,0	0069	LDI R69,0
071	LDI R86,0	070	LDI R70,0	0070	LDI R70,0	0070	LDI R70,0
072	LDI R87,0	071	LDI R71,0	0071	LDI R71,0	0071	LDI R71,0
073	LDI R88,0	072	LDI R72,0	0072	LDI R72,0	0072	LDI R72,0
074	LDI R89,0	073	LDI R73,0	0073	LDI R73,0	0073	LDI R73,0
075	LDI R90,0	074	LDI R74,0	0074	LDI R74,0	0074	LDI R74,0
076	LDI R91,0	075	LDI R75,0	0075	LDI R75,0	0075	LDI R75,0
077	LDI R92,0	076	LDI R76,0	0076	LDI R76,0	0076	LDI R76,0
078	LDI R93,0	077	LDI R77,0	0077	LDI R77,0	0077	LDI R77,0
079	LDI R94,0	078	LDI R78,0	0078	LDI R78,0	0078	LDI R78,0
080	LDI R95,0	079	LDI R79,0	0079	LDI R79,0	0079	LDI R79,0
081	LDI R96,0	080	LDI R80,0	0080	LDI R80,0	0080	LDI R80,0
082	LDI R97,0	081	LDI R81,0	0081	LDI R81,0	0081	LDI R81,0
083	LDI R98,0	082	LDI R82,0	0082	LDI R82,0	0082	LDI R82,0
084	LDI R99,0	083	LDI R83,0	0083	LDI R83,0	0083	LDI R83,0
085	LDI R100,0	084	LDI R84,0	0084	LDI R84,0	0084	LDI R84,0
086	LDI R101,0	085	LDI R85,0	0085	LDI R85,0	0085	LDI R85,0
087	LDI R102,0	086	LDI R86,0	0086	LDI R86,0	0086	LDI R86,0
088	LDI R103,0	087	LDI R87,0	0087	LDI R87,0	0087	LDI R87,0
089	LDI R104,0	088	LDI R88,0	0088	LDI R88,0	0088	LDI R88,0
090	LDI R105,0	089	LDI R89,0	0089	LDI R89,0	0089	LDI R89,0
091	LDI R106,0	090	LDI R90,0	0090	LDI R90,0	0090	LDI R90,0
092	LDI R107,0	091	LDI R91,0	0091	LDI R91,0	0091	LDI R91,0
093	LDI R108,0	092	LDI R92,0	0092	LDI R92,0	0092	LDI R92,0
094	LDI R109,0	093	LDI R93,0	0093	LDI R93,0	0093	LDI R93,0
095	LDI R110,0	094	LDI R94,0	0094	LDI R94,0	0094	LDI R94,0
096	LDI R111,0	095	LDI R95,0	0095	LDI R95,0	0095	LDI R95,0
097	LDI R112,0	096	LDI R96,0	0096	LDI R96,0	0096	LDI R96,0
098	LDI R113,0	097	LDI R97,0	0097	LDI R97,0	0097	LDI R97,0
099	LDI R114,0	098	LDI R98,0	0098	LDI R98,0	0098	LDI R98,0
100	LDI R115,0	099	LDI R99,0	0099	LDI R99,0	0099	LDI R99,0
101	LDI R116,0	100	LDI R100,0	0100	LDI R100,0	0100	LDI R100,0
102	LDI R117,0	101	LDI R101,0	0101	LDI R101,0	0101	LDI R101,0
103	LDI R118,0	102	LDI R102,0	0102	LDI R102,0	0102	LDI R102,0
104	LDI R119,0	103	LDI R103,0	0103	LDI R103,0	0103	LDI R103,0
105	LDI R120,0	104	LDI R104,0	0104	LDI R104,0	0104	LDI R104,0
106	LDI R121,0	105	LDI R105,0	0105	LDI R105,0	0105	LDI R105,0
107	LDI R122,0	106	LDI R106,0	0106	LDI R106,0	0106	LDI R106,0
108	LDI R123,0	107	LDI R107,0	0107	LDI R107,0	0107	LDI R107,0
109	LDI R124,0	108	LDI R108,0	0108	LDI R108,0	0108	LDI R108,0
110	LDI R125,0	109	LDI R109,0	0109	LDI R109,0	0109	LDI R109,0
111	LDI R126,0	110	LDI R110,0	0110	LDI R110,0	0110	LDI R110,0
112	LDI R127,0	111	LDI R111,0	0111	LDI R111,0	0111	LDI R111,0
113	LDI R128,0	112	LDI R112,0	0112	LDI R112,0	0112	LDI R112,0
114	LDI R129,0	113	LDI R113,0	0113	LDI R113,0	0113	LDI R113,0
115	LDI R130,0	114	LDI R114,0	0114	LDI R114,0	0114	LDI R114,0
116	LDI R131,0	115	LDI R115,0	0115	LDI R115,0	0115	LDI R115,0
117	LDI R132,0	116	LDI R116,0	0116	LDI R116,0	0116	LDI R116,0
118	LDI R133,0	117	LDI R117,0	0117	LDI R117,0	0117	LDI R117,0
119	LDI R134,0	118	LDI R118,0	0118	LDI R118,0	0118	LDI R118,0
120	LDI R135,0	119	LDI R119,0	0119	LDI R119,0	0119	LDI R119,0
121	LDI R136,0	120	LDI R120,0	0120	LDI R120,0	0120	LDI R120,0
122	LDI R137,0	121	LDI R121,0	0121	LDI R121,0	0121	LDI R121,0
123	LDI R138,0	122	LDI R122,0	0122	LDI R122,0	0122	LDI R122,0
124	LDI R139,0	123	LDI R123,0	0123	LDI R123,0	0123	LDI R123,0
125	LDI R140,0	124	LDI R124,0	0124	LDI R124,0	0124	LDI R124,0
126	LDI R141,0	125	LDI R125,0	0125	LDI R125,0	0125	LDI R125,0
127	LDI R142,0	126	LDI R126,0	0126	LDI R126,0	0126	LDI R126,0
128	LDI R143,0	127	LDI R127,0	0127	LDI R127,0	0127	LDI R127,0
129	LDI R144,0	128	LDI R128,0	0128	LDI R128,0	0128	LDI R128,0
130	LDI R145,0	129	LDI R129,0	0129	LDI R129,0	0129	LDI R129,0
131	LDI R146,0	130	LDI R130,0	0130	LDI R130,0	0130	LDI R130,0
132	LDI R147,0	131	LDI R131,0	0131	LDI R131,0	0131	LDI R131,0
133	LDI R148,0	132	LDI R132,0	0132	LDI R132,0	0132	LDI R132,0
134	LDI R149,0	133	LDI R133,0	0133	LDI R133,0	0133	LDI R133,0
135	LDI R150,0	134	LDI R134,0	0134	LDI R134,0	0134	LDI R134,0
136	LDI R151,0	135	LDI R135,0	0135	LDI R135,0	0135	LDI R135,0
137	LDI R152,0	136	LDI R136,0	0136	LDI R136,0	0136	LDI R136,0
138	LDI R153,0	137	LDI R137,0	0137	LDI R137,0	0137	LDI R137,0
13							



**Dave Crisp has been looking at serious business software from Gemini — here's his verdict.**

THIS LITTLE OFFERING COMES FROM Gemini, a company purporting to be the home of "serious software". The title suggests that Gemini has more business goodies in the pipeline and the "serious software" slogan bears some thinking about. Looking up "serious" in my dictionary gives the synonyms such as "grave" — perhaps a portent of the consequences of running the software! Anyway, here's the facts:

Business Pack has 1 Gemini's & Cashbook system, a VAT file facility and a Final Accounts package. A pretty comprehensive line of programs for anyone who needs to keep accounts.

The pack comes complete with three manuals/leaf guides. These are optional provided it's a little light on content and obviously devised by someone who knows accountancy and computers a great deal better than he does the user.

My accountant was quite impressed with the facilities offered by the package, but threw up his hands in horror when I explained that it was designed for use by small businesses with little or no computer or other book-keeping or data processing.

The review copy came on disk and loaded perfectly enough, albeit slowly. The tape-based version comes on three cassette cassettes and, presumably, takes long enough to LOAD to allow you to go down to the Dog & Duckette for tea.

Reading a spelling mistake in the opening paragraph of the operating instructions starts doing nothing to improve my confidence, but plough on regardless.

Each screen begins by requesting you to enter a date. Remember, if accounts is vital to the firm, forget the review problem the day of the week for now — for tea time!

Various prompts appear to assist you as the screen progresses. Obviously when using the system up from scratch there are functions to carry out which won't be required again until the start of a new accounting year. The first of these (and potentially the most dangerous) is prompted by "Initiate a new file!". A "Y" response requires confirmation. At this stage if you respond in the affirmative, all account balances are cleared down — not much fun if you are halfway through your financial year!

Cashbook entries are restricted to 25 transactions each, whilst analysis and posting totals are only allowed to five

# BUSINESS FILE

separate accounts. If you are inputting a lot of data it is wise to do some pre-processing analysis of your own (i.e. sort different types of transactions into separate print). Gemini recommends labelling each batch with a header containing details of the entries — imperative if you are to have any effect against the various batch which will eventually appear on your printed reports.



Each of the three systems gives you a menu. The screens obviously vary according to the particular service you wish to access. Every conceivable alternative is included, but be sure you fully understand the implications of the actions you take — particularly at the Journal entry stage. Single entry journals are frowned on by many accountants. To make sure you're it is better to keep plenty of documentary back-up and remember that there is no automatic entry to Control or BAL general accounts. Always enter at the user of caution.

There are enough report options available to satisfy the most critical to accountants and the output from the VAT File system should be sufficient to allow you to complete your returns without any problems.

The usual dangers stalk the night for as problems are concerned — a generic problem. Don't switch off in the wrong sequence or when a program is running. Also, if you use anything except a Commodore printer (who doesn't?) you will need to change the ASCII code for it. The relevant code should appear in your generic handbook.

I would strongly recommend that you purchase one of the "Accounting Made Easy" type books and have a good read before using the package.

And you have some grasp of the principles of double entry book-keeping you will have untold problems. Gemini advises that you approach your accountant prior to preparing profit & loss accounts and balance sheet — seriously, like a good idea.

Two particular aspects of the package are especially poor:

Firstly, paragraphing the screen of the manual referring to data saving if didn't have the nerve to test it! "If an error occurs while you are saving data, type GOTO 540 and try again!" — how polished and professional can you get? When I was a green young programmer if I'd suggested to my project leader that such a statement should appear in the operator's instructions, I would have been fired almost as soon as hired.

Secondly if you accidentally hit RUN/STOP and RESTORE unsuccessfully you automatically lose any data input during the session.

I find it difficult to recommend Commodore's Business Pack too far as there are a number of better and cheaper alternatives. Gemini still has three important lessons to learn. Do more market research, when the "basic" slogan works — convert them to a faster medium — such as magnetic tape; produce good system analysis and a technical author.

Bill on box 2

The best part of this particular issue is the lovely lady who appears, smiling, on the front cover. If only someone would plug her C64 into the TV monitor for her













# COMPETITION COMPETITION

Fancy the chance of winning  
a prize worth £1000?

Enter our competition and  
you could win one of 30

copies of our Game of the  
Month.

## The rules

ENTRIES WILL NOT BE ACCEPTED FROM employees of Angus Specialist Publications, Wizard Development Company Ltd and Alabaster Partners and, therefore, this restriction also applies to employee's families and agents of the companies.

The how to enter section forms part of the rules. The Ed's decision is final and no correspondence will be entered into.

## How to enter

There are a few differences between the two pictures. Study them carefully and circle the differences on the picture attached to the coupon. Fill in the coupon clearly and seal the picture and coupon in an envelope. Write clearly the number of differences you found on the back of the envelope.

Post your entry to:  
William Compton, Your Commodore,  
No 1 Golden Square, London W1R 3AB.  
The closing date is first post on Thursday, 11th October 1985. You may enter as many times as you like but each entry must be on an official coupon - not a copy - and sealed in a separate envelope.

Important: Please follow closely the guidelines on entering - incomplete coupons and entries with no numbers on the back cannot be considered. If you are a winner, the coupon will act as a label for your prize so clear writing is essential.



Name .....  
Address .....  
.....  
.....  
.....  
.....  
Post code .....

Number of differences found .....  
Send for Bill and Compton, Your Commodore, No 1 Golden Square,  
London W1R 3AB. Write clearly and fully and don't forget to put the number  
of differences you found on the back of the envelope.





## U.N. Soft Centre

**Figure 6.** The effect of the number of iterations on the accuracy of the proposed algorithm. The figure shows two bar charts comparing the accuracy of the proposed algorithm against the number of iterations (10, 20, 30, 40, 50) for two different datasets: "Dataset A" and "Dataset B". The y-axis represents Accuracy (%) from 80 to 95. In both cases, accuracy increases with more iterations, peaking at 30-40 iterations before slightly declining or stabilizing.

Dataset	Iterations	Accuracy (%)
Dataset A	10	~87
	20	~90
	30	~92
	40	~91
	50	~90
Dataset B	10	~85
	20	~88
	30	~90
	40	~89
	50	~88



For help with Reading and Writing  
**01-405 4017**



for Literacy  
 Adult Literacy & Basic Skills Unit  
 PO Box 200 London WC1V 2ET

## Electronic Aids (Tewkesbury) Ltd. Accounting and Educational Software

For Commodore 64 and 128

**SALES LEDGER** with invoicing: **£99.00 + VAT**

Invoices to client to your own design, statements, aged invoices, daybook, period and VAT report, label printing etc. etc.

**FINANCIAL ACCOUNTS** **£90.00 + VAT**

Figures, ledger with inputs for sales, purchases, cash and payments, allows you to structure your own reports e.g. profit & loss, balance sheet, department reports etc. Complete and final trial balance, opening balances, last month's trial balance, monthly totals on all accounts etc.

**PURCHASE LEDGER** **£89.00 + VAT**

Sales ledger, purchase ledger and final accounts (as above) all combined into one integrated package.

**INTEGRATED ACCOUNTS** **£199.00 + VAT**

Sales ledger, purchase ledger and final accounts (as above) all combined into one integrated package.

**SEVENPACK** **£48.00 + VAT**

A simple integrated package for very small businesses. Purchase & Sales Control, invoicing, period account ledger, statements, ordinary stock control, stock control for our parts and address label printing.

**PATROLL** with new NI rules: **£99.00 + VAT**

Easy to use, but also has 7 smart independent forms, 4 pre-set adjustments and 100% 8 adjustment adjustments, deduction card, coverage analysis etc.

**ELECTRONIC AIDS LTD.**  
 62 High Street, EYESHAM,  
 Wores. WR11 4HS  
 Phone 0386 - 49339

Money back on any item returned within 28 days.

"Chess game it is not" Your Computer

## THE CHESS GAME

Commodore 64 - Now Available - RRP £7.95

"If Lewis Carroll had written a computer program then this would be it" *Home Computing Weekly*

"Brilliantly designed opening screen. Prospective animation and sprite design are excellent... A very clever and original program." *Commodore Magazine*

"Excellent 3D graphics... Top Notch" *Commodore Computing International*

"I strongly recommend this game whether you understand chess or not." *Home Computer Weekly*

"An arcade extravaganza from MicroClassics." *Your 64*

### THE FIVE STAR REVIEWS CONTINUE...

*Home Computer Weekly* - Commodore Share Products Review Week - The chess game rated top game - *Popular Computing Weekly* - Rated equal top five week - *Commodore Magazine* - Top two best game - top graphics - *Commodore Computing International* - Five star graphics -

### PRIORITY ORDER SERVICE

If your dealer has sold out, simply send us a cheque or postal order for £7.95 and we will rush you a copy by air mail.

MICROCLASSICS, GREENFIELD, PERKY ROAD,  
 BURNLEY, BLANCKLEY



## VIZASTAR 64

Vizastar 64 is a powerful business management system for Commodore 64 and 128. It includes a comprehensive range of modules for sales, purchases, cash and payments, invoicing, statements, aged invoices, daybook, period and VAT report, label printing etc. etc. It also includes a powerful financial accounts module for profit and loss, balance sheet, department reports etc. Complete and final trial balance, opening balances, last month's trial balance, monthly totals on all accounts etc.

**Master 64** is a powerful business management system for Commodore 64 and 128. It includes a comprehensive range of modules for sales, purchases, cash and payments, invoicing, statements, aged invoices, daybook, period and VAT report, label printing etc. etc. It also includes a powerful financial accounts module for profit and loss, balance sheet, department reports etc. Complete and final trial balance, opening balances, last month's trial balance, monthly totals on all accounts etc.

## VIZAWRITE 64

Vizawrite 64 is a powerful business management system for Commodore 64 and 128. It includes a comprehensive range of modules for sales, purchases, cash and payments, invoicing, statements, aged invoices, daybook, period and VAT report, label printing etc. etc. It also includes a powerful financial accounts module for profit and loss, balance sheet, department reports etc. Complete and final trial balance, opening balances, last month's trial balance, monthly totals on all accounts etc.

Module	Price	Price	Price	Price	Price
Vizastar 64	£79.95	£79.95	£79.95	£79.95	£79.95
Master 64	£79.95	£79.95	£79.95	£79.95	£79.95
Vizawrite 64	£79.95	£79.95	£79.95	£79.95	£79.95
Vizastar 64	£79.95	£79.95	£79.95	£79.95	£79.95
Master 64	£79.95	£79.95	£79.95	£79.95	£79.95
Vizawrite 64	£79.95	£79.95	£79.95	£79.95	£79.95

Calco Software

Unit 1204, Hillside, Kingston Hill, Surrey, KT1 1AP, Tel: 01-844-0114







been defined which adds two and seven and prints the result

```
1 ADD 2 7 + .
```

By simply entering ADD, the number zero will be displayed. This is not a very good way to add two numbers, so like Basic, Forth can get the two numbers at run time

```
ADD = .
```

To add two numbers the following is entered:

```
2 5 ADD  
7 OK
```

Notice the 'OK' which is the equivalent of Basic's Ready prompt.

## Reverse Polish Notation

Forth requires that numbers to be operated upon should be on the stack prior to the operator. In other words, to perform a mathematical function, it is necessary to place the operand(s) before the operator(s), and not use them as we humans do.

1 2 + is the equivalent of 1 + 2

We humans perform our calculations in this manner, but Forth does it in postfix notation, known as reverse polish notation. Here are some algebraic expressions in this notation (postfix notation is the reverse of Forth's equivalents)

Normal	Forth
A + B	A B +
A * B	A B *
A * B + C	A B * C +
A * B + C * D	A B * C D * +

It looks confusing at first, but it is surprising how easy it is to adapt.

The reason why postfix notation is used is because computers find it easier. As a result, the also contributes to Forth's speed.

## Definitions

In Basic programs words are written, but in Forth words are defined. To define the word that you are defining a Forth word, two Forth words 'C' and 'D' are used together with a name for the word

```
name forth-words.
```

Once the definition is entered, it is placed within the dictionary and can be used just like any other Forth word. Other Forth words could use other definitions.

One of the great advantages that Forth has over conventional procedure languages is that it is possible with many versions of Forth to define the package that you wish to make your words part of the language



well. Thus, the next time you load Forth your own defined words will be part of the language!

Finally, the following Basic Forth speed comparison demonstrates the speed of Forth. The two programs are identical in operation, and perform the following:

```
Clear the text screen.  
Fill the screen map with the value 1, in other words  
POB2 to the screen display its complete contents.
```

The result is that a computer of 8's will be displayed, then 8's and so on. In BASIC the test took 133 seconds to complete, and in Macintosh Forth it took 10.23 seconds.

```
10 PRN "CLES"  
20 FOR C=1000 TO 10000  
30 POB2 C  
40 NEXT C  
50 FOR I=1 TO 30  
60 FOR J=1000 TO 10000  
70 POB2 J  
80 NEXT J  
90 NEXT I
```

```
1000 CLES  
10000 CLES  
10000 CLES  
2000 CLES  
20000 CLES  
1000 CLES  
10000 CLES  
1000 CLES
```

## Reviews

It is very easy to implement Forth on a micro-especially one like the 68010. As a result, there are at least seven implementations of the language available in the U.K.

As you can see from the list, the prices of Forth packages differ. The features offered also vary, and it is a good idea to check about why you need the package. If, for instance, you wish to expand an existing Forth, with a multi-task package, then Fig Forth could be what you are looking for. On the other hand, while lightning is fast there is always a thunder. Another very good version is Forth from Melbourne House. With this package it is possible to produce your own distribution programs with ease.

Reviewed here are just four of the packages currently available.

**Fig-Forth**  
**Romik Software**  
**Tape only — £14.95**

Although this version is one of the cheapest, and offers no test set or editor it is a great package!

This is because of the way in which the package has been implemented. Romik Forth offers the user a true Fig Forth environment without any additions of any kind. This has advantages. First, it saves recovery time that would otherwise be used for corrections that the user may not want in the first place. And secondly, the



user can expand and customise the package without having to worry about workspace that may be used by others on the same disk.

Thus said, I wouldn't recommend Romek forth for the absolute beginner. Forth is unfriendly by nature, and Romek's version, an exact implementation, reflects that.

In use, the Romek forth proved very fast, even compared with other Forths. The only size aspect was the screen output. I believe the authors have made a C64 to the Atari/Amiga ROM and used the same routines to limit the screen output. This is a pity, as it is possible to optimise the routine, thus speeding it up.

Being an accurate trap translation of Fig-Forth, Romek forth uses the same editor as most Forths - and a horrible editor it is too!

When forth was designed, no method was specified for inputting source code. A US member of the Forth Interest Group (hence Fig) designed a crude but practical line editor, published in the user groups newsletter. The author suggested it should be used until someone designed a better one. Unfortunately, everyone (including the software house) decided to use it, and it's still being used for forth implementations today!

#### Forthold

##### Audiogenic

Cartridge only—£29.95

The standard routine used in Fig-Forth is not the only 'idiot-proof' about the language. Saving and LOADING is and from a backup device is also complicated.

However, in this case, there is a practical reason. Forth has designed to use dynamic storage meaning that memory is treated as the backup device. The idea behind this is to speed-up SAVING and LOADING. Only when you physically save or store or LOAD the data (at the start or end of an editing session perhaps) would you PURGE the data to disk or tape. In other words, dynamic store is a guarantee the memory and tape/disk.

But managing the Atari disk is tricky business in standard Forth. Audiogenic Forth-64 gets round the problem by allowing easy to use fast and reliable saving and loading program screens, but only that but the ROM is fully supported, so it is possible to control (with some care) any device based on the back of the 64.

Audiogenic's Forth is supplied as a cartridge thus SAVING the user the agony of LOADING from tape to disk. Not only that, but bringing a cartridge takes memory for more source screens which is quite important.

Numerous words cover tape-to-disk I/O within the 64 line, but there is no graphic area and extensions on the 64 version. But it should be noted that the 64-25 version

does have a word for changing the border and screen colour as well as loading the saved screens.

The editor used in this version is based around the original, but makes good use of the Commodore function/arrow keys. What is a bit unusual is that the screen and the direct mode is a forth editing screen. This means that when the end of the screen is reached, it is necessary to clear it before continuing. This is quite awkward, and takes away some of the immediacy of direct execution.

However, screen (as in editing) management is very well explained & a lot of the mundane tasks being done automatically. A screen is selected, and after the program source has been saved, it can be loaded into main memory and run. To actually save at least a number of source screens, a file is specified in the SAVE/SAVE command, and forth will save or load the required number of blocks - single!

#### White Lightning

##### Qash Software

Tape—£19.95, disk—£29.95

White Lightning is quite simply the best value for money you can get. For the price, you not only get a good implementation of Fig-Forth, but you also get a graphics development system.

White Lightning is a fairly complex package that can be used to develop commercial programs. The package is forth based and includes a Fig-Forth routine. Built around this is a graphics development shell called GDBA, which contains numerous screen and sound words which are used in the forth program. As well as this, it is possible to write basic within a White Lightning or machine code. Once the program has been developed and debugged, it is possible to save it as a stand alone version that can be freely distributed.

Both disk and tape versions are available, but the 'horror' programmer should opt for the disk version as it is necessary to use and load source data as well as the forth source if a graphics program is being developed.

As mentioned it is possible to include basic statements within the forth source. This will mean that the program will not run as fast as it would in forth only, but it is a great help when starting off with the language.

#### FORBEE

##### Melbourne House

Tape—£14.95

Although only recently released, Forth has a long and interesting history. The Fig-Forth based package was initially developed for the ZX Spectrum by Abbotsoft.

Forth is supplied on cassette which is recorded in the Rastrow format and takes under three minutes to load. Disk users will be relieved to hear that the package can be transferred to disk with just two simple commands. Once saved to disk, the package takes 40 seconds to load and uses the disk as the main backing device.

Supplied with the cassette is a groovy 16 page manual which covers the bare essentials of the package. There are a number of features within the package that are not documented in the manual, but I understand that Melbourne House will be issuing a full length manual in the near future.

Forth is based around Fig-Forth and offers a number of interesting extensions. First, the screen editor has been adapted to make use of the C64's editor. Source code can be entered into any of the nine screens in a free format, and an edit session can be aborted if desired. One point missing from the editor is the ability to delete carriage returns. This may not sound important, but when editing in a free format (such as the example program) the becomes essential.

Another interesting feature is the ability to produce stand-alone programs with the package. After your masterpiece is created, DAP followed by the name of the word will save a machine code file which can be 'fired from basic. However, the code that is saved is as large as the forth itself (38 blocks), and, probably, contains most of the package. There are more elegant methods of achieving the same thing.

Screen modes included within the package are:

MONA Mode colour mode  
MONA Normal colour mode  
FGBA Extended background colour mode  
HGBA Normal background colour mode  
HGB High-res mode  
LRS Low-res mode

All the natural colour functions such as border, ink and background are supported, together with plot and draw. Screen setup commands are included as well as 'to save' commands.

In use, Forth proved to be excellent. Even a two-fail crash could be recovered from re-trying back into the main code. The package is not overladen with features, but the features which it does have are perfectly adequate.

#### Other Forth Packages

There are numerous versions of FORTH available, and here is a list of some:  
Forth-64 Header Software £34.95  
Forth-64 Hardware £74.95  
Fig-Forth Audiogenic £19.95, £24.95





Nick Hampshire brings you a  
detailed look at the C-16  
Operating System variables  
and memory map.

# NUTS & BOLTS OF THE C/16

ANYONE THINKING TO DO EXTENSIVE machine code programming on the C-16 or Plus/1 will require two essential pieces of information.

The first is a list of the operating system variables, their location and function. This is essential if total interaction with the system software is not to occur. It is as important to know where to place variables when using any of the system subroutines.

The second essential piece of information is a memory map. This informs the entry points to the operating system and all its ROM routines. The use of such routines within a program can save a considerable amount of time and program space, an important consideration with the C-16.

I have included both these pieces of information in this article.

Unfortunately, Commodore has not previously produced any of this data and it has therefore been necessary to use the experience gained with previous Commodore machines to deduce the location and function of both variables and routines. All the locations discussed have been thoroughly checked and I have no reason to doubt that they are correct. I was, of course, very interested to hear from any readers with additional information.

The operating system variables are, in common with all Commodore machines, stored in the bottom area of memory extending from location \$0000 up to the bottom of the screen colour memory at \$8000, a total of 3K of memory. All the important variable locations are shown in Table 1. It should be noted that locations \$00 and \$01 are as in 0 port located on the processor chip and cannot therefore be used for variable storage.

A memory map of the ROM based operating system and Basic routines is shown in Table 2. Many of these routines are usable to some degree, and some are extremely valuable. Most of the routines and variables are similar, or even identical in function, if not in location, to those on the C64.

Readers wishing to gain all its depth knowledge of the routines, functions and operations are recommended to consult one of the advanced books on the list. The 64 Extended series by Nick Hampshire, published by Cordon is recommended, especially *Advanced Commodore 64 Basic Remodel*.

Table 1

## Operating system variable storage.

\$001	- Cassette control
Bit 4	: Cassette read
3	: Cassette motor (0=on)
1	: Cassette write
\$008	- 0=LOAD, 1=VERIFY
\$009	- Type IF=string, 00=numeric
\$00E	- Type: 80=integer, 88=floatline
\$00F	- DATA scan/LIST mode/memory flag
\$014-\$015	- Integer value
\$016	- Pointer: temporary string stack
\$017-\$018	- Last two string vector
\$019-\$021	- Temporary string stack
\$022-\$025	- Utility pointer area
\$026-\$029	- Product area for multiplication
\$02B-\$02C	- Pointer: start of Basic
\$02D-\$02E	- Pointer: start of Basic variables
\$02F-\$038	- Pointer: start of arrays
\$031-\$032	- Pointer: end of arrays
\$033-\$034	- Pointer: bottom of strings
\$035-\$036	- Pointer: current string
\$037-\$038	- Pointer: top of Basic memory
\$039-\$03A	- Current Basic line number
\$03B-\$03C	- CHARACTER pointer
\$03D-\$03E	- Pointer: FOR/DO stack
\$03F-\$040	- Current DATA line number
\$041-\$042	- Current DATA address
\$043-\$044	- Input vector
\$045-\$046	- Current variable name
\$047-\$048	- Current variable address
\$049-\$04A	- Variable counter for FOR/NEXT
\$04B-\$04C	- V variable surveillance pointer name
\$04D	- Comparison symbol accumulator
\$04E-\$053	- Misc numeric work area
\$054-\$056	- Jump vector for functions
\$057-\$058	- Misc work area
\$051	- FAC#1 exponent
\$062-\$065	- FAC#1 mantissa
\$06C	- FAC#1 sign
\$067	- Serial evaluation constant pointer



## Table (continued)

0068	- FRC01 overflow
0069-006E	- FRC02
006F	- FRC sign comparison
0070	- FRC01 rounding
0075	- Rope for graphics screen (00not area label)
007C-007D	- Pointer G0508 stack
0083	- File for window (040>window on, 400=multicolour, F00=both)
0088	- Status word ST
0091	- Keyboard C1A-STOP and RYS flag
0093	- Onload.verify
0094	- Serial output/deferred char flag
0095	- Serial deferred character
0097	- # open files
0098	- Input device
0099	- Default CMD device
009A	- Direct8080/run80 output control
009D-009E	- Tape end address/End of error
00A2-00A5	- Jiffy clock
00A6	- Serial bit count/E01 flag
00A8	- Countdown tape write/bit count
00A9	- # chars in filename
00AC	- Current logical file
00AD	- Current secondary address
00AE	- Current device
00AF-00B0	- Pointer to filename
00B2-00B3	- I/O start address
00B4-00B5	- Alt start address (load/verify)
00B6-00B7	- Pointer/loadcassette buffer
00C4-00C5	- Input cursor loc (row,column)
00C6	- Which key-64th key
00C7	- Input from screen/keyboard
00C9-00C9	- Pointer to screen line
00CA	- Pointer/cursor column
00CB	- Output quotes flag
00CD	- Pointer/cursor row
00CE	- Output character (to screen)
00CF	- # of indents outstanding
00EA-00EB	- Screen colour pointer
00EC-00ED	- Keyboard pointer
00EF	- # chars in keyboard buffer
00F0	- Type of tape file
00FF-0100	- Floating to ASCII work area
0100-010E	- Tape error loc
0100-01FF	- Processor stack area
0200-0250	- Basic input buffer
0250-025A	- Pointer/line# for CNT
025B-025C	- Pointer/Basic statement for CNT
02F2-02F3	- Float-fixed vector
02F4-02F5	- Fixed-float vector
0300-0311	- Basic vectors





Table 1(continued)

0312-0313	- I/O vector for keyboard/clock
0314-0315	- Main I/O vector for sound duration/graphics split
0316-0317	- IRQ interrupt vector
0318-0319	- OPEN vector
031A-031B	- CLOSE vector
031C-031D	- Set input vector
031E-031F	- Set output vector
0320-0321	- Restore I/O vector
0322-0323	- INPUT vector
0324-0325	- Output vector
0326-0327	- Test-STOP vector
0328-0329	- GET vector
032A-032B	- Abort I/O vector
032C-032D	- User vector
032E-032F	- LOAD vector
0330-0331	- SAVE vector
0332-03F2	- Cassette buffer
0470	- CHRGET subroutine
04FC/04FE	- Duration for voice 1
04FD/04FF	- Duration for voice 2/noise
0500	- RND seed value
0500-0512	- Logical file table
0513-051C	- Device # table
051D-0526	- Secondary address table
0527-0530	- Keyboard buffer
0531-0532	- Start of usable memory
0533-0534	- End of usable memory
0535	- Serial bus timeout flag
0538	- Current colour code
Bit 7 = 1=flash	
6-4 = luminance (0-7)	
3-0 = colour (0-15)	
053F	- Maximum size of keyboard buffer
0541	- Repeat speed counter
0542	- Repeat delay counter
0543	- Keyboard shift/control flag
0544	- Case switch count
0545-0546	- Keyboard table setup pointer
0547	- Case switch disable
0700-0700	- Bomb attack
07F2	- SVS R reg save
07F3	- SVS X reg save
07F4	- SVS Y reg save
07F5	- SVS status reg save
07F6	- Last key
07FD	- Countdown for double TI bus
0800-08FF	- Colour memory
0C00-0CFF	- Screen memory
1000-3FFF	- Basic program memory
1800-3FFF	- Graphics screen/colour memory
37F6-37FD	- Reset entry (when ROM is out)





Table 2

## Commodore 16 Memory Map.

0000	- Basic ROM cold start routine	9000	- Constant = 32768
0019	- Initialize Basic	9005	- Float-fixed
0021	- Powerup message	9006	- Perform 'PPE'
0105	- Vectors for #0000	9070	- Perform 'F08'
0117	- Initialize vectors	9080	- Perform 'DEP'
0123	- CHASET for #0479	90DE	- Perform 'FN'
018C	- Keywords	90E6	- Perform 'STRs'
0383	- Command vectors	9C40	- Get temporary string
0415	- Function vectors	9C80	- Perform 'CHRS'
0453	- Arithmetic operator vectors	9CCE	- Perform 'LDTS'
0471	- Error messages	9D03	- Perform 'RIGHTs'
0681	- 'SYNTAX ERROR'	9D15	- Perform 'RIDE'
0683	- Error routine	9D61	- Perform 'LDX'
0703	- 'READY.'	9D70	- Perform 'PSC'
070F	- Basic warm start	9D84	- Get 1 byte parameter
083D	- Find Basic line	9D8D	- Perform 'VRL'
0A79	- Perform 'H04'	9D92	- Get parameters for POKE/INKEY
0A90	- Perform 'CLR'	9DFA	- Perform 'PDEE'
0AF1	- Set test counter	9E12	- Perform 'POKE'
0AFF	- Perform 'LIST'	9E6A	- Perform 'WAIT'
0B8C	- Perform 'SKIP'	9E87	- Perform 'substract'
0C9A	- Perform 'RESTORE'	9E9C	- Perform 'add'
0CD0	- Perform 'STOP'	9F0E	- Perform 'LDR'
0CDA	- Perform 'END'	AF70	- Perform 'waitfor'
0D03	- Perform 'CONT'	AF37	- Perform 'divide'
0D2C	- Perform 'DISJUN'	AF21	- Memory to PAGE1
0D43	- Perform 'GOTO'	AF21	- PAGE2 to PAGE1
0D83	- Perform 'RETURN'	AF20	- Round PAGE1
0D80	- Perform 'DATA'	AFBE	- Perform 'SHR'
0DE1	- Perform 'IF'	AFD0	- Perform 'RRR'
0E00	- Perform 'REDY' and 'ELSE'	AF50	- Perform 'BIT'
0E1B	- Perform 'DN'	AFD4	- Perform 'SGR'
0E3E	- Set line number (0-63999)	AFDE	- Perform 'space'
0E7C	- Perform 'LET'	AFD7	- Perform 'newate'
0F00	- Perform 'PRINT4'	AF60	- Perform 'ESP'
0F0C	- Perform 'CHR'	AF87	- Perform 'RRR'
0F00	- Perform 'PRINT'	AF70	- Basic I/O error handler
0F00	- Print string from (u,a)	AF55	- Basic-kernal match for OPEN
0F00	- Perform 'SET'	AF53	- Basic-kernal match for PRINT
0F0E	- Perform 'INPUT4'	AF51	- Basic-kernal match for INPUT
0F00	- Perform 'INPUT'	AF57	- Basic-kernal match for set output device
0F14	- Perform 'READ'	AF56	- Basic-kernal match for set input device
0F03	- 'READY FROM START' message	AF5F	- Basic-kernal match for GET
0F04	- Perform 'NEXT'	AF55	- Perform 'SHR'
0F0C	- Evaluate expression	AF5E	- Perform 'SHR'
0F30	- Constant = 01	AF5B	- Perform 'VCRIFY'
0F45	- Perform 'NOT'	AF73	- Perform 'LDR'
0F45	- Scan word '?'	AF40	- Perform 'OPEN'
0F45	- Scan word 'C'	AF58	- Perform 'CLOSE'
0F41	- Scan word '.'	AF6B	- Get parameters for LOAD/SAVE
0F70	- Perform 'ON'	AF60	- Get parameters for OPEN/CLOSE
0F7B	- Perform 'AND'	AF64	- Setback collect
0F20	- Perform comparisons (<=)	AF70	- Perform 'OSR'
0F3B	- Perform 'DIR'	AF77	- Perform 'STR'
0F40	- Locate variable	AF6B	- Perform 'TRN'
		AF10	- Perform 'RTN'



Table 2 (continued)

FE80	- Perform 'NUMBER'	D065	- Inwait until carriage return
FE80	- Perform 'FOR'	EB11	- Read keyboard
FE90	- Perform 'DELETE'	ED49	- Output to screen
FE90	- Get name for LIST/DELETE	EB1E	- Keyboard select vectors
FE2E	- Perform 'TRAP'	EB2E	- Unshifted table
FE40	- Perform 'RESUME'	EB67	- Shifted table
FE44	- Perform 'PUDEF'	EB68	- CBN low table
FE57	- Perform 'DO'	EB69	- Control table
FE9C	- Perform 'EXIT'	EL29	- Shift/run equivalent
FE83	- Perform 'LOOP'	EB6A	- Delay for 0.5 sec
FE92	- Perform 'TROH'	EBCC	- Find any time header
FE55	- Perform 'TROH'	EB21	- Find a specific header
FE6D	- Perform 'AUTO'	EB09	- Get (#B320)
FE6D	- Perform 'HELP'	EB28	- Inwait (#B322)
FE79	- Perform 'KEY'	EB4B	- Output (#B324)
FE49	- Perform 'SOUND'	EB08	- Receive from serial
FE89	- Perform 'VOL'	EB3F	- Send serial deferred
FE01	- Perform 'PRINT'	FB18	- Set input device (#B31C)
FE04	- Perform 'CMR'	FB08	- Set output device (#B31E)
FE02	- Perform 'BOX'	EBFA	- Send 'talk'
FE05	- Perform 'CIRCLE'	FE1A	- Send talk SR
FE29	- Perform 'SQUARE'	EE2C	- Send 'listen'
FE29	- Perform 'SQUARE'	EE4D	- Send listen SR
FE1E	- Perform 'CIRCLE'	FE5D	- Close file (#B31A)
C429	- Perform 'DRAW'	CEB8	- Find file entry
CEB0	- Perform 'LOCATE'	CEFD	- Get file details
C51A	- Perform 'COLOR'	CE80	- Abort all files (#B32B)
CE67	- Perform 'SCHEDULE'	EF0C	- Restore default I/O (#B329)
CEB0	- Perform 'SCALE'	EF23	- Send 'unlisten'
CEC9	- Perform 'GRAPHIC'	EF3E	- Send 'untalk'
CE8C	- Perform 'DIRECTORY'	CE55	- Open file (#B31B)
CE41	- Perform 'DRIVE'	F06A	- Load programs (#B32E)
CE51	- Perform 'DLOGS'	F1B4	- Save programs (#B32F)
CE68	- Perform 'HEADER'	F265	- Test STOP low (#B32E)
CE9C	- Perform 'SCRATCH'	F2B4	- Power reset entry
CE9C	- Perform 'COLLECT'	F30C	- Set kernel vectors
CE0A	- Perform 'COPY'	F309	- Store kernel vectors
CEF4	- Perform 'RENAME'	F308	- Initialize I/O
CE60	- Perform 'BACKUP'	F309	- Initialize system constants
CE21	- Get parameters for disk commands	F48C	- Set filename details
CE0F	- Read disk error	F413	- Set file details
CE02	- "ARE YOU SURE?" messages	F41B	- File status
CE80	- IRQ/SRK entry	F41C	- Get status
CEDE	- IRQ routine (#B314) - handles graphics wait, sound duration	F42D	- Get timeout
CE42	- IRQ routine (#B312) - handles clock, keyboard	F427	- Read/set top of memory
CECD	- Handle sound durations	F42D	- Read top of memory
CEFD	- Run clock	F42F	- Set top of memory
CF26	- Get time	F436	- Read/set bottom of memory
CF2D	- Set time	F438	- Read bottom of memory
CE80	- Character definitions (256)	F43E	- Set bottom of memory
FE02	- Screen address low	F445	- Monitor call entry
FE02	- Screen address high	F44C	- Monitor SRK entry (#B316)
FE04	- Get screen size		User vector (#B32C)
FE09	- Put/set row/column	FE15	- Get I/O address
FE08	- Set screen pointers	FE32	- IRQ entry
FE01	- Rescue char from keyboard	FE3E	- IRQ exit
		F008	- TED memory
		FE32	- Perform 'MONITOR'





This month A.P. and D.J.

Stephenson provide the key

to Basic and delve into the

mysteries of keywords.

# T.H.E. BASIC F.A.C.T.S.

SOME READERS WHO HAVE BEEN following this series may have found the last few articles a little tough. This month, we thought it would be fun to pause for breath and go over some of the lesser used BASIC keywords in detail.

The Basic language, as implemented on the C64, employs a variety of keywords which are sufficient to cope with most situations. A good proportion of the total are in constant use and have been treated and used in earlier articles in this series. For various reasons we have neglected some of them altogether. A few others have been used without adequate description. Although all keywords are defined in the user manual, alternative treatments can often lead to a better understanding.

The vocabulary of any language is seldom confined to the list indexed, in everyday speech, we only use a fraction of the total number of words we know. It is the same with programmers. However rich the Basic vocabulary, it is easy to get into a rut by relying only to those keywords which are easy to understand or use. Unfortunately, the most serious adviser of the C64 or VIC II would be happy to advise that the Basic vocabulary in these machines is, to say the least, sparse. Because of this, it is particularly important that we know how to make efficient use of every keyword. Before dealing with them individually, it goes to clarify them into order.

## Types of keyword

A keyword is any combination of characters, chosen for their mnemonic value, recognised by the interpreter as an order to be carried out. They fall into one of three classes - statements or functions.

Statements are keywords which have an overall effect on the complete program. They are more often employed outside a program in direct mode. However, they may also occur within a program under a line number. The following keywords are direct commands: GOTO, UNT, GOSUB, RETURN, SAVE, VERIFY.

Statements are keywords which perform some particular action within a program. Most keywords are statements.

Functions are specialised statements which perform a standardised operation on a variable. Functions can be

recognised by the brackets which introduce the variables. The following complete list of Commodore functions uses X or Y in example numbers: variables and A4 for string variables. ATN(X), CHR(X), COS(X), EXP(X), FRE(X), INT(X), LEFT(X), LEN(X), LOG(X), MID(X), MIN(X), MOD(X), POS(X), RIGHT(X), RND(X), SIN(X), SPC(X), SQR(X), STR(X), TAN(X), VAL(X), VAR(X), VARI(X).

IF and THEN are exceptional in that Commodore displays them as functions although they do not require brackets round the variables. Numeric values used in functions are often called arguments. The arguments in two of the functions, FOR and NEXT, are clauses, meaning that some number must be entered to avoid triggering a syntax error although it doesn't matter what number you choose within the range 1 to 32767.

## The command RUN

RUN clears all variables when it starts a program from the first line number. It is possible to start a program from lower down the program by using GOTO X, where X is the starting line number. RUN will then lead to error conditions because variables created under previous line numbers are cleared from memory.

If you want to restart the program a second time at some lower point, it is better to use GOTO X which leaves previously created variables unchanged. The main danger with using GOTO is triggering a "NO DATA IN AREA" error if a GOTO line is re-executed.

## The command CONT

A program comes to a halt under any one of the following conditions:

- The RUN/STOP key has been pressed.

- The program has reached either a STOP or END statement.
- The program has reached the last line number.
- The program has encountered a standard error condition.

The program can always be restarted from the beginning or, by typing RUN X, started from line X. However, RUN or RUN X always clears variables from memory which were created by the last run. Only the program is retained.

The command CONT can provide a powerful weapon for stalling out long a program during the development stage. It is usually possible, providing the development has proceeded in accordance with guidelines of good structure, to recognise certain critical points in a program.

To check the value of variables at these critical points, temporary STOP instructions can be inserted. When the program is RUN, it will halt at the first STOP and you can check the state of the variables by printing out their value. If the values are different to what you expected then a break point is proceeding with the rest and the reason for the discrepancy has been found. Assuming that everything is OK, you can allow the program to carry on with the next segment by again using CONT.

Unlike RUN, which always starts the program cleanly, CONT allows it to carry on from where it was stopped and preserves the value of all variables reached up to that point. These temporary STOPs act as "break points" in the program, enabling the user to be reorganised and faults remedied in a series of easy steps.

To take full advantage of break points, the program should be fixed, easy to use, where the results of calculations can easily be checked manually at the end of each one. Remember that CONT will not



work. If the program has halted due to an error condition, if you try it, the response will be "CAN'T CONTINUE."

## The command NEW

As far as Basic is concerned, the command NEW will clear the entire memory, including any pre-existing programs. Although Basic is usually entered in direct mode, you may include it within the program under a line number. If you start on a new program where the previous one or its elements are still in memory, be prepared for some unexpected results. Switching OFF and ON again will clear everything but entering BASIC is more elegant.

A thoughtful designer can take the tedious of repeatedly switching the power on and off more than absolutely necessary. Repeated use of the main switch may even reduce the life of some of the more sensitive chips.

## The command LIST

This is probably the most overworked command in Basic. Although the default listing is to the screen, a previously active CMD statement can be used to output the listing to a printer or other variable peripheral.

You may include LIST within a program under a line number but, after the listing, the program will always return to command mode and the READY message. Unless it is placed at the line immediately preceding the END statement, it is difficult to imagine what purpose LIST can serve within a program other than saving you the trouble of entering LIST afterwards (but then you don't always want a listing after every run).

## The command SAVE

This command can only be used to save programs. It saves on tape by default or on disc by quoting drive number right after the program name. The Commodore method of saving on tape is clever in that some other machines, partly because two copies are always involved that need error-free playback are easily garbled by degradation between the versions.

If SAVE is used within a program under a line number, the program is halted while the tape or disk is being recorded. On completion of the save operation, it comes in with the rest of the program. In this way, it is possible to save a program and run it in one go, although you'll probably never need to do this.

## The LOAD command

When you load a program using LOAD as a direct command, all variables are cleared from memory including any

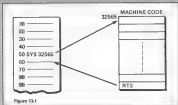


Figure 10.1

existing program which may be there. If LOAD is used as a statement within a program, although the existing program is cleared, the running variables are untouched and a BASIC automatically follows. Used in this way, the LOAD command acts as a "chain" command so that several programs on tape or disk can be chained together. This allows large programs which would exceed the bounds of memory, to be entered in parts small enough to be accommodated within the memory space available. Each subsequence program finds an environment already prepared by previous programs. The individual subprograms must be carefully designed to ensure that, apart from their set of running variables, they are self-sufficient.

For example, one program cannot GOTO a line number of another because it would not be in the computer's memory. The program bytes are normally loaded into a block of memory starting at address 3240 (3800 hex) onwards.

## The VERIFY command

Programs use VERIFY immediately after they have saved a program on tape. It takes a long while to write a large program and just as long to verify it afterwards so it is a precaution which many programmers neglect to take. In any case, the most likely reason for failing to VERIFY is an error during playback rather than during record.

However, VERIFY does have one additional advantage. Because it loads the tape several inches after the program's end it is safe to remove and the program immediately without worrying about overlap.

## POKE and PEAK

These two statements are best considered together because they occupy the same, midway between Basic and the inner

workings of the machine. In traditional Basic, the individual p-gram holes in memory are of no consequence because the interpreter takes care of all memory assignments.

POKE allows us to have a look at the byte currently occupying memory address 0. To read an "BASIC QUANTITY" message, it must be within the range 0 to 255, and it must be a positive number.

POKE X,Y allows us to store the number X in address Y. Since a memory location can only hold one byte of data, Y must be within the range 0 to 255, and it must be a positive number.

To PEAK a number can never be harmful but an indiscriminate POKE can turn sour. Poking the right number into the wrong address or vice versa can crash a program. The results of a crash vary but it's usually either a scowling of dancing screen characters or a noisy indifference to all keyboard activity. The crash will almost certainly occur if a number is poked into one of the workspace areas used by the operating system.

You should regard the operating system with respect as it's capable of withholding all cooperation if crossed.

Providing care is taken in choosing memory addresses, POKE can be used to:

- Assign data bytes directly. For example, if you need to store numbers less than 256 in memory, it can be done far more economically by using POKE than by assigning the number to a Basic variable. It is not so convenient or flexible as normal assignments but worth bearing in mind when a lot of small numbers have to be manipulated.

- At one of the elements in graphic work.
- Passing parameters to machine code subroutines called from within Basic.
- In spite of the warning regarding a POKE into operating system areas, there are a number of special



locations which can be **POINT** in order to modify certain effects.

- 14. To control colour. **POINT** **S100** (hex) set background colour to orange.
- 15. It is possible to operate on selected bits of a memory byte by using the **AND** or **OR** operators in conjunction with **POKE** and **PEEK**. For example, let 4 in location **S1020** decide whether the multiplayer character made 5 or 6 (as 4=0) or 0 (as 4=1). To ensure the off condition, use

**POKE S1020,(PEEK(S1020) AND 255)**

This may seem a little obscure unless you are familiar with the binary features of **AND**.

## The AND statement

It should not be difficult to understand the meaning of **AND** and **OR** when used in conjunction with the **IF/THEN** structure. For example

**100 IF A < B AND C = D THEN PRINT X**

The order is almost self explanatory and clearly states that both conditions must be true. However, there are other, less obvious, operators lurking beneath the surface which are known as *bitwise operators*. Study the following series of operators:

Assume the pattern 11070911  
how **AND** it with 11001115  
The result is 11000005

Note that the result is the same as the original pattern except where the **AND** pattern had a 0 in that position. The **AND** pattern, known as a *mask*, is calculated thus: Use '0' in the mask when you want 0 in the result, otherwise use '1'.

Let's try a practical example. Suppose a certain memory location holds the following bit pattern:

10011000

Let's also suppose that we wish to clear bit 4 or 4 without disturbing the others, (bits are numbered 0 to 7, from right to left so bit 4 is the fifth bit from the right). The correct **AND** mask will be 11111115. Unfortunately, there is no provision in **GBA Basic** for entering for patterns direct. We have to use decimal so we must convert the bit pattern to the **AND** mask to decimal before it can be accepted. The above example required an **AND** mask of 9991111.

Working laboriously in decimal, this becomes  $1 \times 2^{10} + 0 \times 2^9 + 1 \times 2^8 + 1 \times 2^7 + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 215$ . **POKE** Commanders don't provide hex numbers - this would make **AND** mask calculations dead easy.

We are done in a position to return to the problem we left in the last column. You will remember that the following line

was supposed to clear bit 4 in location **A1000** to 0:

**POKE S1020,(PEEK(S1020) AND 255)**

We are getting the same bit pattern back into the location after we have **AND**ed it with 255. We know that 255 decimal is 11111111 which is the correct **AND** mask for clearing bit 4 in the original location. To compensate, we make that the following apparent algorithm: we calculate  $255 - 2^{(4 \text{ AND } 251 = 251, 7 \text{ AND } 8 = 8)}$

## The OR statement

This, like the **AND**, has binary connotations. As we have seen, **AND**ing a condition when we want to clear certain bits to 0 but **OR**ing is used when we want to set certain bits to 1. The rule for the **OR** mask is a follows. Use '1' in the mask where bits are to be '1's, otherwise use '0's.

Suppose we start with 11001011  
Then we **OR** it with 00010000  
The result is 11011011

Note that the result is the same as the original except in the position where there was a 1 in the **OR** mask.

Let's return to our previous example once again but this time, assume we want bit 4 in location **S1020** to be a 1. The **OR** mask must be 00010000, which in decimal is 16. Our **POKE** line then becomes:

**POKE S1020,(PEEK(S1020) OR 16)**

To summarise, contains the following: **POKE** **S1020** = 7, 5 **OR** 2 = 7

## The NOT statement

this is the third member of the trio capable of bitwise operations. Before describing its uses, we must first define a few terms: To flip, means to change a 1 to 0 or vice versa. The bit complement (logical complement or one's complement) of the bit pattern formed by flipping all the bits. The two's complement is the same as the bit complement except that an extra 1 is added (example

Starting with 10010111  
Bit complement is 01101000  
Two's complement is 01101001

The **NOT** statement forms the two's complement of a bit pattern which can lead to rather mysterious results. For example, **NOT 1 = 255**. To see why, note that  $1 = 00000001$  so the two's complement is  $11111110 + 1 = 255$ .

The mystic, clear up what we learn that a negative number, is far as the computer is concerned, a really the two's complemented form of the equivalent positive. When we **NOT** something, we are really asking for the negative version

At this information on the bitwise operations of **AND**, **OR** and **NOT** is really crossing the border between Basic and machine code. The same thing can be said regarding the next item.

## The SYS command

It is possible to mix Basic with machine code in the one program. There are several situations where such a mix may be justified. The execution speed of Basic is sufficient for some purposes but intolerable for others. Another disadvantage of Basic is that memory is not used economically. For example, in Basic, the memory is usable for numbers is fixed, irrespective of their magnitude. It takes just as many bytes to store the number three as it does to store 9999. Machine code programs can arrange for memory storage to be more closely tailored to magnitude.

However, machine code is not everyone's idea of the good life so, as far as possible, most of the program may continue to be written in the tranquil environment of Basic, with only the occasional leap into and out of machine code. Figure 11.1 shows how such leaps can be performed using the **KEYWORD** **SYS**.

Note from the figure that we can only jump to a machine code sub routine if we happen to know where it is located in the memory map. In other words, we must know the starting address. If you've written the machine code there's no problem but you may want to use some of the ready made routines already residing in the ROM. There are two, providing you find out the starting addresses, consulting the Programmer's Reference Guide or employing some truly detective work. Remember you can always **PEEK** locations in ROM.

There is one obstacle that can arise in the use of **SYS**. Many subroutines only work if information is provided. To quote a simple example, a subroutine to find the square of a number obviously can't work unless it is supplied with the number to be squared. When such a subroutine is written, it is assumed the numbers to be already residing at a particular memory location. The address will form part of the accompanying documentation.

Therefore, before we go on the sub routine by means of **SYS**, we must pass this information in the required address. These typical call might look like this:

100 **POKE** 1620 16  
110 **SYS** 34500

The data location is 1620 and the number 16 has been passed into it. The subroutine is assumed to start at address 34500. The machine code sub routine must end with **RTS** (return from Subroutine), otherwise control will not come back to Basic.





SPOTS  
BEFORE  
FOUR  
THIRTS



```

100 REM          DOIMAGES
110 REM BY BOY TICHARD 1985
120 GOSUB230
130 :
140 FORM=1TO15:READFM:MS=MS+CHR
50(M):NEXTFM
150 DATA5,85,32,82,75,85,32,84,75
,84,87,75,85,82,68
160 POKES280,0:POKES281,0
170 :
180 PRINT"(CLEAR)(DOWN)(DOWN)(DOWN
3".FORM-5TO3:PRINTTAB(15)DOWN);:N
EXT
190 :
200 PRINT"(DOWN)(DOWN)(DOWN)(DOWN)
",MS
210 :
220 PRINT"(DOWN)(DOWN)(DOWN)(DOWN)
(DOWN)(DOWN)(YELLOW) (RUSH) DO
YOU NEED IN
INSTRUCTIONS (RUSHFF) (Y-N)"
230 GETA:IFAB="":THEN230
240 IFAB="M":THEN120
250 IFAB="Y":THEN260
260 GOTO230
270 :
280 PRINT"(CLEAR)(GREEN)
RANDOM INSTRUCTIONS (RUSHFF)"
290 :
300 PRINT"(DOWN)(DOWN)(NORMAL DOWN
0 RULES APPLY....."
310 :
320 PRINT"(DOWN)YOU WILL BE PLAYIN
G WITH A DOUBLE SIX"
330 :
340 PRINT"(DOWN)PACK AND DEAL MIN
E DOMINOS."
350 :
360 PRINT"(DOWN)THE FIRST 'DROP' W
ILL BE RANDOM EITHER"
370 :
380 PRINT"(DOWN)YOU OR THE COMPUT
E DOING FIRST."
390 :
400 PRINT"(DOWN)IF YOU DAIN 1ST OR
OF THEN YOU CHOOSE"
410 :
420 PRINT"(DOWN)YOUR DOMINO BY NUM
BER (1 TO 31"
430 :
440 PRINT"(DOWN)THE DOING IS THEN
DISPLAYED IN THE"
450 :
460 PRINT"(DOWN)CENTRE OF THE SCRE
EN."
470 :
480 PRINT"(DOWN)(DOWN) (YELLOW
MICRO) PRESS RETURN TO CONTINUE
(RUSHFF)"
490 GETA:IFAB="":THEN500
500 IFAB>CHR(13)THEN500
510 :
520 PRINT"(CLEAR)(DOWN)THE COMPUT
E WILL AUTOMATICALLY MAKE"
530 :
540 PRINT"(DOWN)ITS MOVE AND RECOR
D THE MOVE AT THE "
550 :
560 PRINT"(DOWN)TOP RIGHT OF THE S
CREEN."
570

```



```

580 PRINT"(DOWN)THE CENTRE DORING
WILL THEN CHANGE TO"
590 :
600 PRINT"(DOWN)SHOW THE MOVE PLAY
ED (I.E) YOU DROP"
610 :
620 PRINT"(DOWN)DOUBLE BLANK THE C
OMPUTER PLAYS ONE"
630 :
640 PRINT"(DOWN)BLANK SO THE CENTR
E DORING CHANGES TO"
650 :
660 PRINT"(DOWN)BLANK ONE, AND SO
ON."
670 :
680 PRINT"(DOWN)A RECORD OF YOUR M
OVES IS KEPT ON THE"
690 :
700 PRINT"(DOWN)BOTTOM RIGHT OF TH
E SCREEN."
710 :
720 PRINT"(DOWN) (WHITE)JASON)
PRESS RETURN TO CONTINUE (RUSOFF)
"
730 GETA$(I$A$="THEN730
740 I$A$<>(CHR(13))THEN730
750 :
760 PRINT"(CLEAR)(DOWN)13IF YOU
A MOVE COULD ALTER THE STATE OF"
770 :
780 PRINT"(DOWN)PLAY (I.E) CENTRE
DORING READS ONE-TWO"
790 :
800 PRINT"(DOWN)AND YOU PLAY ONE-T
WO THEN YOU WILL BE"
810 :
820 PRINT"(DOWN)OFFERED THE CHOICE
OF MAKING IT EITHER"
830 :
840 PRINT"(DOWN)ONE UP OR TWO UP (
BOTH ENDS THE SAME)"
850 :
860 PRINT"(DOWN)YOU THEN PRESS FRO
M 'O'(BLANK) TO 'S"
870 :
880 PRINT"(DOWN)DEPENDING ON WHICH
DORING YOU CHOOSE."
890 :
900 PRINT"(DOWN)WHITE(JASON) IF
YOU CANNOT PLAY A DORING THEN (RUS
OFF)"
910 :
920 PRINT"(DOWN) (RUSONS P
RESS RETURN (RUSOFF)C= 13"
930 :
940 PRINT"(DOWN)YOUR MOVE WILL THE
N BE RECORDED AS (YELLOW)(RUSON)DU
T(RUSOFF)(C)"
950 :
960 :
970 PRINT"(DOWN)(DOWN) (CYAN)
(RUSONS PRESS RETURN TO CONTINUE (
RUSOFF)"
980 GETA$(I$A$="THEN980
990 I$A$<>(CHR(13))THEN980
1000 :
1010 PRINT"(CLEAR)(DOWN)(DOWN)(CYEL
LOW)IF A GAME IS 'STITCHED' (NEITH
ER PLAYER"
1020 :
1030 PRINT"(DOWN)(CAN SO) THEN THE
WINNER IS THE PLAYER"
1040 :
1050 PRINT"(DOWN)WITH THE LEAST NU
MBER OF SPOTS."
1060 :
1070 PRINT"(DOWN)(DOWN)THE DOMINCE
S ARE COLOUR CO-ORDINATED"
1080 :
1090 PRINT"(DOWN)FOR EASY IDENTIFI
CATION."
1100 :
1110 PRINT"(DOWN)(DOWN)(RED) ON
(EYELLOW) IS RED : TWO IS YELLOW "
1120 :
1130 PRINT"(DOWN)(GREEN) THREE(
YELLOW) IS GREEN (CYAN)FOUR(YELLO
W) IS CYAN"
1140 :
1150 PRINT"(DOWN)(PURPLE) FIVE(
YELLOW) IS PURPLE (C= 13)SIX(YELLO
W) IS ORANGE"
1160 :
1170 PRINT"(DOWN)(DOWN) (C= 63)
(RUSONS PRESS RETURN FOR FIRST GAME
(RUSOFF)"
1180 GETA$(I$A$="THEN1180
1190 I$A$<>(CHR(13))THEN1180
1200 :
1210 PRINT"(CLEAR)":COSUBSS30
1220 SUB(0)="BLANK "
1230 SUB(1)="ONE "
1240 SUB(2)="TWO "
1250 SUB(3)="THREE "
1260 SUB(4)="FOUR "
1270 SUB(5)="FIVE "
1280 SUB(6)="SIX "
1290 :
1300 SUB="(RUSON)(YELLOW)OUT(RUSOF
F)(GREEN)"
1310 :
1320 SUB(1)="(YELLOW)(NONE)(DOWN)(D
OWN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)
(DOWN)(DOWN)"
1330 :

```













```

10DOWNC:DOWN
11DOWNC:DOWNC:DOWNC:DOWNC1"-GOSUB37
20:GOSUB5100,RS(8)=""
2500 IFV1=1THENL1=L2:GOSUB5110:GO
SUB5100:GOSUB3800:GOTO3140
2610 IFV2=1THENL1=L1:GOSUB5110:GO
SUB5110:GOSUB3800:GOTO3140
2520 IFV2=2THENR1=R1:GOSUB5110:GO
SUB5110:GOSUB3800:GOTO3140
2530 IFV1=2THENR1=R2:GOSUB5110:GO
SUB5110:GOSUB3800:GOTO3140
2640 :
2650 :
2660 PRINT"YELLOWCHROME:DOWNC:DO
WN:C:DOWNC:DOWNC:DOWNC:DOWNC:DOWNC:
DOWNC:DOWNC"
:T=16:GOSUB5180
2670 IFV1=1THEN3020
2680 FORA=1TO1
2690 JP=V1:GOSUB510
3000 PRINTTAB(T:SPB,;T-T+2,IFT>16T
HENPRINT,T=16
3010 NEXTA
3020 T=20
3030 :
3040 PRINT"YELLOWCHROME:DOWNC:DO
WN:C:DOWNC:DOWNC:DOWNC:DOWNC:DOWNC:DOWNC:
DOWNC:DOWNC"
3050 IFV2=1THEN3100
3060 FOR2=1TO2
3070 JP=V2:GOSUB510
3080 PRINTTAB(T:SPB,;T-T+2,IFT>20T
HENPRINT,T=20
3090 NEXT2
3100 LD=V1:RD=V2:ND=1:GOSUB5180:GO
SUB5110
3110 :
3120 K=10:ES="CHROME:DOWNC:DOWNC:
DOWNC:DOWNC:DOWNC:DOWNC:DOWNC:DOWNC:
DOWNC:DOWNC:DOWNC:DOWNC:DOWNC:DOWNC:
DOWNC:DOWNC:DOWNC:DOWNC:DOWNC:DOWNC:
DOWNC:DOWNC"
3130 GOSUB37
20,RS(8)="",GOSUB5180:GOTO3140
3140 :
3150 :
3160 :
3170 FORA=1TO5
3180 IFX1(A)=""THEN3190
3190 C1(A)=VAL(LEFT$(X1(A),1))
3200 C2(A)=VAL(RIGHT$(X1(A),1))
3210 IFC1(A)=1THENL1=L2:C2(A)=GOTO33
0
3220 IFC1(A)=2THENR1=R2:C2(A)=GOTO33
0
3230 IFC2(A)=1THENL1=C1(A):GOTO33
0
3240 IFC2(A)=2THENR1=C1(A):GOTO33
0
3250 NEXTA

```

```

3260 :
3270 RS(1)=RS(3)+"CHROME:LEFTC:LE
FTC:LEFTC"+"CR:DOWNC:YELLOW:DOWNC:V
GOF:DOWNC1"
3280 :
3290 PRINT"CHROME:DOWNC1"TAB(37)RS
(1):GOSUB510
3300 FORDELAY=1TO500:NEXTDELAY
3310 IF2=1THEN3300
3320 GOTO2650
3330 GOSUB5110
3340 :
3350 RS(1)=RS(1)+"CHROME:LEFTC:LE
FTC:LEFTC"+"CR(A)
3360 :
3370 PRINT"CHROME:DOWNC1"TAB(37)RS
(1)
3380 IFA=1THEN7=0
3390 IFA=2THEN7=4
3400 IFA=3THEN7=6
3410 IFA=4THEN7=12
3420 IFA=5THEN7=16
3430 IFA=6THEN7=20
3440 IFA=7THEN7=24
3450 IFA=8THEN7=28
3460 IFA=9THEN7=32
3470 X1(A)=""
3480 :
3490 Z2=0:K=0:ES="CHROME":GOSUB37
50:GOSUB5180:GOSUB510:GOSUB3800:GO
SUB510
3500 :
3510 PRINT"YELLOWCHROME:DOWNC:DO
WN:C:DOWNC:DOWNC:DOWNC:DOWNC:DOWNC:
DOWNC"
3520 FORA=1TO5:PRINTTAB(15)RS(A),
NEXTA
3530 :
3540 PRINT"YELLOWCHROME:DOWNC:DO
WN:C:DOWNC:DOWNC:DOWNC:DOWNC:DOWNC:
DOWNC:DOWNC"
:T=16
3550 IFL1=1THEN3510
3560 FORA=1TO5
3570 JP=L1:GOSUB510
3580 PRINTTAB(T:SPB,;T-T+2,IFT>16T
HENPRINT,T=16
3590 NEXTA
3600 :
3610 PRINT"YELLOWCHROME:DOWNC:DO
WN:C:DOWNC:DOWNC:DOWNC:DOWNC:DOWNC:
DOWNC:DOWNC"
:T=20
3620 IFR1=1THEN3580
3630 FORA=1TO3
3640 JP=RD:GOSUB510
3650 PRINTTAB(T:SPB,;T-T+2,IFT>20T
HENPRINT,T=20
3660 NEXTA

```





[illegible]



```

4510 IFND=8THEN=40
4520 IFND=7THEN=44
4530 IFND=6THEN=48
4540 IFND=5THEN=32
4550 RETURN
4560 C1=0,C2=0,V1=0,V2=0
4570 :
4580 PRINT"CLEAR?";FORA=0TO9:PRINT
TAB(10)DOWN;:NEXTA
4590 :
4600 PRINT"(DOWN)"TAB(12)"WHITE3S
TITCHED GAME"
4610 :
4620 PRINT"HOME"(DOWN)(DOWN)(DOWN
)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DO
WN)(DOWN)(DO
WN)(DOWN)(DOWN)(DOWN)(YELLOW)"TAB(
30)"TIME"
4630 FORA=1TO9
4640 IFX(A)=""THEN=4660
4650 C1=C1+VAL(LEFT(X(A),1))
4660 C2=C2+VAL(RIGHT(X(A),1))
4670 PRINTAB(35)X(A)
4680 NEXTA
4690 :
4700 PRINT"HOME"(DOWN)(DOWN)(DOWN
)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DO
WN)(DOWN)(DO
WN)(DOWN)(DOWN)(DOWN)(YELLOW)YOURB
[= 7]"
4710 FORA=1TO9
4720 IFX(A)=""THEN=4750
4730 U1=U1+VAL(LEFT(X(A),1))
4740 U2=U2+VAL(RIGHT(X(A),1))
4750 PRINTAB(0)X(A)
4760 NEXTA
4770 :
4780 PRINT"YELLOW"(HOME)(DOWN)(DO
WN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DO
WN)(DOWN)(DO
WN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DO
WN)"TAB(26)"1 ":"PRINTAB(25)"HAW
E";PRINTAB(25)C1-C2"
4790 :
4800 PRINT"YELLOW"(HOME)(DOWN)(DO
WN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DO
WN)(DOWN)(DO
WN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DO
WN)"TAB(10)"YOU ":"PRINTAB(10)"H
AVE";PRINTAB(10)V1+U2"
4810 :
4820 IFC1=C2+U1+U2THENPRINT:PRINT:
PRINTAB(15)"I WIN":GOTO4830
4830 :
4840 IFU1+U2+C1+C2THENPRINT:PRINT:
PRINTAB(15)"YOU WIN":GOTO4830
4850 :
4860 IFU1+V2+C1+C2THENPRINT:PRINT:
PRINTAB(15)"WE DREW":GOTO4830
4870 :
4880 [VU]=LOWAND2-RTDTHDVS10

```



```

4890 IFU1=2DANDV2=LDTHDMS10
4900 GOTO4920
4910 IFLD=RTDTHDMS20
4920 :
4930 D0SUB5740,G0SUB5740,D0SUB5740
4940 PRINT"YELLOW"(HOME)(DOWN)(DO
WN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DO
WN)(DOWN)(DO
WN)"
TAB(0)SLK(LO):PRINTAB(0)"UP OR "
4950 PRINTAB(0)SUB(2):PRINTAB(0)
"LF"
4960 D0SUB5740,G0SUB5740,D0SUB5740
4970 GETSDS:IFSDS=""THEN=4970
4980 SD=VAL(SDS)
4990 IFSD+LDANDSD>80+80+80
5000 G0SUB5740,G0SUB5740,G0SUB5740
5010 :
5020 PRINT"YELLOW"(HOME)(DOWN)(DO
WN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DO
WN)(DOWN)(DO
WN)"
TAB(0)" ":"PRINTAB(0)"
PRINTAB(0)"
5070 PRINTAB(0)" ":"PRINTAB
(0)"
5080 :
5090 R=10,DDS="HOME"(DOWN)(DOWN)(DO
WN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN
)(DOWN)(DOWN
)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DO
WN)"
5090 IFSD+V1THDSD+V1,LD=V1,G0SUB5
110,G0SUB5110,G0SUB5330,G0SUB3700,
MS(8)="",GDS
L83830
5070 IFSD=V1THENGOTO3140
5080 IFSD=V2THENLD=U2,RD=U2,G0SUB5
110,G0SUB5110,G0SUB5330,G0SUB3700,
MS(8)="",GDS
L83830
5090 IFSD=V2THENGOTO3140
5100 GOTO4910
5110 VL=LD+UR+RD
5120 :
5130 PRINT"YELLOW"(HOME)(DOWN)(DO
WN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DO
WN)(DOWN)(DO
WN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DO
WN)"TAB(0)SUB(VL)
5140 :
5150 PRINT"YELLOW"(HOME)(DOWN)(DO
WN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DO
WN)(DOWN)(DO
WN)(DOWN)(DOWN)(DOWN)(DOWN)(DOWN)(DO
WN)"TAB(26)SLK(VR)
5160 RETURN
5170 :
5180 :
5190 SS=54272,LF=55,WF=55+1,AD=55+
■
5200 SS=55+5,WF=55+4,UD=55+24
5210 H3=55+15,L3=55+14
5220 FORX=55TO55+24

```









Our resident expert  
answers your  
programming queries.

## INPUT

Most computers with function keys or "macro" keys have been trying to do that for some time on my C64 but without too much success. Could you possibly tell me how this could be done? *Armando Doyle Cleveland*

## OUTPUT

Unfortunately, as you have no doubt found out, there is no simple way to define the C64. Most Basic extension packages (see our August and September issues) have a command which allows you to program keys by means of a new keyword, it is however quite simple to use these keys in your own programs without having to go to any expense. Take a look at the following short program:

```
10 GOTO 45
20 IF A$ = CHR$(10) THEN 15
30 IF A$ = CHR$(13) THEN 15
40 A=ASC(A$)-127-128-A-1
50 PRINT "FUNCTION KEY  
"A" IS REMOVED"  
60 GOTO 10
```

This short routine will scan for all eight function keys. By adding the following line it is also possible to add four function keys by using the Commodore key as an extra shift key.

```
45 IF FLAG=0 THEN  
A=A/24
```

Inserting the function keys in this manner is extremely simple as each one of the keys returns a unique number to the GOTO instruction in line 15.

Below is a program which will allow you to actually store strings of characters on each key, the strings can either be keywords or strings of your own choosing.

Where re-definable keys are returned for and these are obtained by pressing F1/F2/F3

# INPUT

and F7 with no shift, the Commodore key as the control key.

The whole program is based on interrupts and can be switched on by:

```
505 49103
```

and off by:

```
505 49102
```

When the program is first loaded the keys are set to:

```
F1 575 49116  
F2 575 49116  
F3 5124 + RTTLR8  
F4 PORG  
F5 1257  
F6 PORG  
F7 1257 + RTTLR8  
F8 49116  
F9 49116  
F10 50715  
F11 50715  
F12 50715  
F13 50440  
F14 50715  
F15 50440  
F16 50440
```

The function of each key can be changed by either pressing F1 or typing 575 49116 followed by the text in quotation marks, a comma and then the key number. The text stored on each key can be no more than 16 characters and the key number no greater than 16, 17 and 12 cannot be changed since they hold the important 575 calls.

Here are a few examples:

```
505 49102 "F1=RTTLR8",7
```

This will PORG 50581 with 6.

```
575 49102 "50440"=CHR$(34)  
"5"=CHR$(34)+5-1
```

This will type 1040 "5" and execute a return.

If you wish to add a RTTLR8 onto the end of any function you must put a left hand arrow at the last character in the text when defining the function.

Below you will find the BASIC source file and a basic loader.

## Machine code

```
5000 *****  
5010 ***  
5020 *** DEFINE FUNCTION KEYS ***  
5030 ***  
5040 *****  
5050  
5060  
5070 " " = 49016  
5080 12502 = 50013  
5090 50716 = 127  
5100 KEYBUF = 521  
5110 KEYVLC = 126  
5120 12502 = 768  
5130 PRINT = 50480  
5140 50481 = 553  
5150 50511 = 11157  
5160 50512 = 50426  
5170 50513 = 50014  
5180 50514 = 50448  
5190 50515 = 48748  
5200 50516 = 20  
5210 52102 = 76  
5220  
5230  
5240 J=PTR J= 12000  
5250 J=PTR J= 12007  
5260 J=PTR J= KEYBUF  
5270  
5280  
5290 12004 SET 12502 (SHIFT ON MBL  
5300 12004 12502 (END OF KEYS)  
5310 12502  
5320 12504 12502  
5330 12504 12502  
5340 12504 12502  
5350  
5360  
5370  
5380 12007 SET 12502 (SHIFT OFF  
5390 12504 12502 (MBL END AND  
5400 12504 12502 (REPLACE THE  
5410 12504 12502 (OLD KEYS)  
5420 12504 12502  
5430 12504  
5440  
5450  
5460  
5470 END 12504 12502 (CHECK TO SEE  
5480 12504 12502 (IF KEY HAS
```





## Machine code

1780	BNE DR	, REPEATED	2040	FOUND0	LDX #0	, PUT DATA INTO
1790	JMP INQOUT		2050		LDY TEMP4	KEYBOARD
1800	LDX DR		2060	KEYCKP	LDX DATA, Y	BUFFER...
1810	BNE NOKEY	, CHECK FOR TWO	2070		CMP #1	, FIND " " AND
1820	CMF #1	, KEYS PRESSED	2080		BNE NOKEYH	REPLACE WITH
1830	BNE NOKEY		2090		LDX #13	CRCLRN
1840	CMF #2		2100	NOKEYH	STA KEYFL, X	
1850	BNE NOKEY		2110		INY	
1860	CMF #3		2120		INY	
1870	BNE NOKEYH		2130		CMF #10	
1880	JMP INQOUT	, FOUND MORE	2140		BNE NOKEY	
1890		, TEMP ONE KEY	2150		STA KEYFL	, SET QJOLE 1%.
1900		, IS PRESSED	2160		JMP INQOUT	, ALL DONE
1910			2170	KEYOFF	LDX DATA, Y	, ALL STRINGS
1920	NOKEYH	, SCORE ALL THE	2180		JSR BACD05	, FROM DATA
1930		, INFO	2190		CMF #11	, CHECK KEY
1940	LDX #3		2200		BCC BACD06	
1950	STA STRFL		2210	STRFL0	LDX #0	, PRINT ERROR
1960	CMF #3		2220	NOKEYH	LDX TEMP0, Y	
1970	BNE NOKEYH		2230		BNE DONT	
1980	LDX #7		2240		JSR BACD06	
1990	NOTD05		2250		INY	
1700	BNE #4		2260		JMP NOKEYH	
1710	STA TEMP2		2270	DONT	STA	
1720	LDX KEYFL0		2280	NOKEY	STA TEMP3	, TIDIE LDA
1730	CMF #3		2290		LDX #0	, RESET THREE
1740	BNE DONT		2300		LDX #0	
1750	JMP INQOUT		2310	BACD05	STA TEMPKEY, Y	
1760	CMF #7		2320		BCC	
1770	BCC DONT		2330		CMF #10	
1780	JMP INQOUT		2340		BCC BACD001	
1790			2350		LDX #0	, STORE STRING
1800			2360	BACD06	LDX CSTRN05, Y	
1810	LDX #0	, FIND THE TEXT	2370		STA TEMPKEY, Y	
1820	STA TEMP4	, FOR BACK KEY	2380		CMF	
1830	LDX TEMP4	, PREPARED	2390		CMF TEMP4	
1840	LDX TEMP4	, ADD 10 TO THE	2400		BCC BACD07	
1850	CLC	, POINTER	2410		JSR BACD0C	, SET NUMBER
1860	ADC #10	, AS, 10 CHAR	2420		JSR BACD0C	, INFLC
1870	STA TEMP4	, FOR KEY	2430		JSR BACD0C	, SET KEY NO.
1880	BCC		2440		LDX NUMBER	
1890	JMP NOKEYH		2450		CMF #7	
1900	FOLWEX	LDX TEMP1	2460		BCC BACD0F	
1910	NOKEYH		2470		BCC	
1920	LDX TEMP1	, ADD #0 TO THE	2480		BCC #0	
1930	CLC	, POINTER	2490		CMF #10	, CHECK #0
1940	ADC #0	, AS, NO CHARS	2500		BCC BACD0F	
1950	STA TEMP4	, FOR FOLW EX	2510	EXD05	JMP SYNTAX	
1960	BCC	, END 1 SHIFT	2520	RTN4, Y	CMF #7	, FIND WORD
1970	JMP NOKEYH		2530		BCC NOKEYH	, TO P/L
1980			2540		INY	, INFO FOR
1990		, TEXT #0, FOUND	2550		LDX DATA, Y	KEY
2000			2560	NOKEYH	INY	
2010			2570		LDX #0	
2020		, FULL BUFFER WITH TEXT	2580		STA TEMP4	



# OUTPUT



## Maritime code (continued)

[illegible][illegible]



## Basic Loader continued

```

1000 DATA78,8,192,78,22,192,78,172
,192,1007
1010 DATA120,162,36,141,20,3,169,1
52,141,230
1020 DATA21,3,88,88,120,163,45,141
,20,707
1030 DATA3,169,234,141,21,3,88,96,
169,920
1040 DATA157,205,43,123,209,3,76,4
9,234,1808
1050 DATA173,141,2,240,17,201,1,24
0,13,1023
1060 DATA201,2,240,9,201,4,240,3,7
6,376
1070 DATA98,234,169,3,141,45,193,1
66,197,1198
1080 DATA141,43,193,201,3,209,2,18
9,7,927
1090 DATA58,233,4,141,44,193,165,1
87,201,1234
1100 DATA5,176,3,76,49,234,201,7,1
41,823
1110 DATA3,76,49,234,169,0,141,46,
193,911
1120 DATA174,44,193,240,13,173,48,
193,24,1100
1130 DATA105,10,141,46,193,205,76,
111,132,1076
1140 DATA174,45,193,240,13,173,46,
193,24,1101
1150 DATA105,49,141,46,193,202,76,
123,199,1124
1160 DATA162,0,172,46,193,196,48,1
93,201,1200
1170 DATA95,208,2,169,13,167,119,2
,800,205
1180 DATA232,224,10,144,236,134,19
8,76,43,1305
1190 DATA234,32,159,173,32,163,182
,201,11,1108
1200 DATA144,15,162,0,169,224,193,
240,7,1174
1210 DATA32,210,255,232,76,184,182
,95,141,1410
1220 DATA17,193,162,0,169,0,157,20
8,193,1123
1230 DATA232,224,10,144,248,160,0,
177,34,1229
1240 DATA159,208,193,200,224,17,19

```

```

9,144,245,1697
1250 DATA32,253,174,32,138,173,32,
247,183,1254
1260 DATA183,20,201,3,144,7,56,233
,1,800
1270 DATA201,16,144,3,76,182,182,2
01,7,1022
1280 DATA176,4,170,193,216,193,170
,169,0,1297
1290 DATA141,47,193,224,0,240,13,1
73,47,1078
1300 DATA193,24,106,10,141,47,189,
202,76,931
1310 DATA8,193,162,0,172,47,193,19
9,208,1173
1320 DATA193,193,48,193,232,200,22
4,10,144,1297
1330 DATA244,56,64,60,0,10,70,83,9
3,716
1340 DATA83,52,67,48,53,56,32,0,82
,464
1350 DATA85,76,96,0,0,0,0,0,258
1360 DATA76,73,83,84,0,0,0,0,316
1370 DATA0,76,73,83,84,36,0,0,0,41
1
1380 DATA0,0,83,83,83,52,67,48,53,
466
1390 DATA53,0,0,63,80,83,62,76,40,
448
1400 DATA0,0,0,0,71,73,84,79,0,313
1410 DATA0,0,0,0,70,82,83,80,298
1420 DATA0,0,0,0,0,77,73,88,218
1430 DATA36,40,0,0,0,0,76,88,221
1440 DATA70,84,36,40,0,0,0,0,82,31
2
1450 DATA73,71,72,84,36,40,0,0,0,3
76
1460 DATA76,65,79,40,0,0,0,0,263
1470 DATA0,53,61,60,52,56,0,0,0,26
2
1480 DATA0,0,53,52,50,55,50,0,0,26
0
1490 DATA0,0,0,76,79,66,68,32,0,37
0
1500 DATA0,0,0,0,83,66,66,63,32,33
5
1510 DATA0,0,0,0,70,82,83,68,293
1520 DATA0,0,0,0,0,0,3,5,2,8
1530 DATA8,3,7,88,82,82,79,82,32,4
42
1540 DATA73,78,32,73,78,80,85,84,0
,582

```


**OUTPUT**




# Software Spotlight

Here's your chance to get some help from the experts on how to spend your hard earned pocket money.

## Frankie Comes to Hollywood

★★★★  
Ocean  
1988  
C64

FRANKIE COMES TO HOLLYWOOD, or anywhere else for that matter, packaged in a smart box with a live live recording of the band performing live for song. What so you can enjoy while you play your next move in the game.

As programs go, this has a lot to offer and the inclusion of games within games reminds me a little of Terminal's Larry Jones but this has all the attributes and tells that Jones lacks. It is streets ahead of any opposition and that's not just because it's set amongst the houses of Mundanoville.

The idea is to reach the title screen, a typically canastary Frankie-style idea, to do this you have to visit all the houses in every street of Mundanoville to find objects which will help you to complete the main games. Really the things you keep in your closet! Old fish, pleasure go to, games and video games.

Talking of videos, there's a vital part in the story. Place a cassette in a home video machine and it reveals a portion of one of the main games. You are then free to enter the screen... Just I'm going into the plot.

Why do you have to enter the main game? That is the lardish purpose behind it all! Before you can enter the final screen you must complete your personality. For the purpose of the game, personality is composed of four facets: pleasure, war, love and faith. Each facet is represented by a symbol as per a hand side of the superb 3D graphics screen and each pleasure pill looks like the symbol which it creates. As the game progresses you gain points in one or more of the

screens for successful results and lose points for failure in the same games. These shown by growing and shrinking columns above the relevant symbols. Points are also awarded for various levels in the houses. Occasionally, Frankie will interrupt with a comment which tells you how your personality is developing as a percentage just to cheer you along.

One of the rooms has a locked door which is the entrance to the Cathedral of Power forming the Pleasure Dome. This is where the main games are located and once you have unlocked the door, entered the maze and fully developed your personality the final door will be found here, but be careful the maze of corridors is full of traps for the unwary.

In one of the houses there has been a murder and once you have discovered the body you are given clues to the murderer's identity and you must not forget that room until you have discovered who the felon is. Each time you play the game you will find a different corpse and a new killer.

I will not reveal the nature of the main games except to say that there are more than 70. Ocean claims that there are 60 problems to solve in all and I believe it. After hours of searching and battling the best rating I achieved was 60%. Personally, in my opinion the game should rank alongside the best C64 titles and every owner should have it in their collection. p.c.



## Lords of Haave

★★★★  
Microdeal  
1988  
C64 — joystick optional

A MOUNT, SCORPIONS AND SPIDERS, A ghostly graveyard and goblins, knights in castles and chests are some of the things you will encounter in the Lords of Haave.

Jedior (the player), a roper in the shape of a man, has come to save the land of Haave from the Dark Lords. His first quest is to seek the Book of Change hidden deep in the village. The Book will reveal the whereabouts of the chest which contains directions to the Dark Lords.

Jedior will have to fight and kill to collect the means to protect himself, such as a Teflon from the Mystics and a key from the Garden, before he can enter the Hall in the Wild Wood and destroy the Dark Lords. If he stays in the Wild Wood too long he will become goblinised

a end of game! Only after many hours of play will the final secrets of the Dark Lords be revealed.

This is most type game as all the follow the same. You can map out the maze when had in sequence (2000 combinations). Half the fun is finding the aim of the game and the other half is arranging the cards in the correct sequence at the beginning of each game that happens very often, until you become experienced. The bottom right-hand corner of the screen has the initials of each card in the game they are set out on the table. The reinforced character set makes them almost indecipherable and the pattern is a relief as two of the cards have the same final.

Once defeated the player is informed of his percentage through the game before being included in the Hall of Fame of his hero. The game leads in under three minutes and has high quality graphics. There are eight levels of play, each with an open/hidden option. The music is exciting and should be turned off immediately using the F1 key

## Triangle

★★★★  
Argus Press Software  
1988  
C64 — disk only

WITH THE AVAILABILITY OF cheap printers, so-called business software is being bought increasingly for home use and here we have a fully integrated word processor, database and spreadsheet package at an extremely low price.

The program is very comprehensive and easy to use, although the standard "manual" provided with it would be woefully inadequate to anybody unfamiliar with the program involved. This is a shame, as the software is unbeatable in terms of pure value for money.

The wordprocessor uses the standard all column screen, with words being broken off at the end of lines as you type. It is completed at the printing stage, but it does mean that you cannot see the finished article on the screen before







## Here and There with the Mr. Men

\*\*\*  
Name: Mr. Men  
\$1.95  
CIB

HERE WE HAVE A SLICE OF POLAR PRISM, game intended to give four to eight year olds practice in distinguishing left, right, up and down - a concept young children find difficult. An element of route-planning is also involved. The popular Mr. Men are used throughout and the graphics, though simple, are very effective.

In Mr. Tickle's game, inside the L, D, L and R keys (or the four arrows) are used to move downwards in the slots of a box with three are lined up with pieces of a jigsaw puzzle which stand outside. When the alignment is correct the pieces move and build up a picture of Mr. Tickle. The jigsaw pieces are very small and are not line up centrally with the doorways, unlike

printing. Special commands are aimed especially and easily using the CTR key, and most of them work very well. A slight problem is that only whole lines of text can be highlighted and moved, but I did not find this a major drawback. This also, was a bit quirky at times.

Part of the machine's memory is set aside as a "clipboard" or workpage, in that chunks of text can be moved at will, and there is an option for printing globally, so that whole documents may be taken together in sequence. A wide range of printing options is provided, including centering and justification. These are accessed through the familiar "format" icons. There are also built-in commands for underlining and italic printing, available on lines and similar printers, though these are implemented in the manual.

The database is about the easiest to use that I have seen, though it is somewhat limited. The program will not sort records into a alphabetical or numerical order. But it is possible to issue through prompts, unless you set aside one of the 17 pre-set lists as a column, and just to

game becomes harder than need be. Chances when I asked to try it were found a test run.

Mr. Tickle and Mr. Grumpy are the other hand, is excellent and quite the best of the four games. Mr. Grumpy is made the best with the space four doorways and Mr. Tickle's extending arm needs to reach in to tickle him. To achieve this, the doorways can be moved as before, then Mr. Tickle must be programmed with a sequence of instructions for moving his arm along the required route. On later levels, chains appear in the doorways, adding to the challenge. If preferred, the directions for movement can be given individually.

In Mr. Lumpy's case of standing with or right is introduced as a puzzle a worm up a tree to a teddy bear. If you are successful he will drop the apple into the mouth of Mr. Lumpy, who is weeping below. Another good game.

Mr. Men versus Mr. Tickle is the poorest of the games in that it is almost impossible to lose. You control four Mr. Men, moving on a thought board, who try to trip Mr. Tickle. It could be quite good, but unfortunately Mr. Tickle's movements are often rather stupid and he rarely wins.

Taken as a whole, the program is fairly good, though rather general. It is certainly not up to Bandwits's usual standard of educational games.

P.B.B.

access the records. Nevertheless, as a simple address book or similar database it is very good and has a particularly suitable procedure for print formatting.

The spreadsheet, or financial planning section of the program is excellent, although it is here that the novice would find some difficulty in following the manual. A rather involved system of relative addressing is used when entering formulas, and the explanation given is very thin indeed.

All the sections of the package are fully integrated, so database and spreadsheet files can readily be incorporated into wordprocessor documents. If you run into difficulty, a "help" key is provided, which puts an explanatory window on the screen. The information given, however, usually just duplicates the inadequate text of the manual.

With proper documentation, tutorial and sample files this package would warrant five stars. As it stands, it is still very good, especially when you consider that it is only a quarter of the price of comparable software. Well done, Apple! P.B.B.

## Cave Fighter

\*\*\*  
Babbage Box  
\$4.95  
CIB - (worth it on keyboard)

I HAD TO BORROW TWO CASSETTE players before I finally managed to get the game to load - and then only from one side! It's a slow loader (very) so there doesn't seem to be any real reason for this. However, I succeeded in the end, so here goes!

You are deep in the heart of a system of caves, from which you are trying desperately to escape before your energy is exhausted. The caves are related web-like, with various, sometimes fatal, and are filled with what look like sharpened spears, contact with which spells instant death! Fortunately you have more lives like a cat, but I guarantee that you will need every one of them.

To escape you will need to run, jump, climb up ropes and shoot arrows - in fact, it is a fairly standard platform game, with a scrolling screen display. The controls are rather unusual, as you jump only after you release the fire button - the height varies according to the length of time you hold the button down. Shooting presents no problems, as you fire automatically in whichever direction you are moving.

There are eight levels, with six caves on each level, and getting through is really challenging!

The graphics are excellent, though the colours are rather muted. Some of the caves are especially good and some very smooth, almost like spiders. The info page displays the three highest scores and gives the option of retaining at the level reached in the previous game. It is accompanied by a rendering of the Star-Spangled Banner in very erratic script. Generally, the sound effects are poor - I missed out turning the volume right down.

The game is quite good, but doesn't really shine in any way, and I quit tired of it.

P.B.B.





# Software Spotlight

## Morden's Quest

\*\*\*  
Melbourne House  
1575  
C64

I'M A "BLAST THE GRIEF OUT OF ANYTHING THAT MOVES" addict, so when I was asked to review this adventure I thought the intergalactic battle was over; by you can imagine that I found Morden's Quest with some trepidation, especially as there wasn't even the slightest hint of any graphics. It was in fact more than pleasant surprise.

Morden's Quest, by Melbourne House, has been written by Peter Joubert you dare give my phone number to anyone! Morden, who also wrote Classic Adventure.

It is rated by the publisher to be "an extremely complex text adventure". They were most right about it being text but they were definitely being underhanded by using the word "complex". I could think up much more graphic superlatives to describe its twists, turns and generally dastardly tricks and absolutely mind-bending puzzles. As I said, I could be much more graphic in my description, but this is supposed to be a family show!

Like any other text game the computer acts as your eyes, ears and hands and is only limited by the 100 words which it understands which, when you come to think about it, is an awfully large vocabulary. However, this has been accomplished by using that well known "text compression" system allowing the author to be very descriptive as you travel through the 150 locations which have been cleverly mapped to look like... Just that would be telling. Typing in "help" will point you in the right direction.

The game starts with you, as the possible answer to my jargon-filled wizard's prayer, falling out of bed. Your quest is to save the universe, as if you hadn't guessed already.

On your journey you will come across evil wizards, mischievous nudgets who are heavily into magic mushrooms, and lean chain clad, human apes, with a strange whine of lament, to name but a few. All you've got to do is collect several parts of a machine which will defeat the pomp and hubris-Boudier who is bent on immortality at the expense of our total destruction.

All this is delivered in such a way that you could be forgiven for believing that you're actually there. I tell you not, this game with its brilliant descriptions of sequences, events, and locations has got to be a must for any software library, even if you think your store is "it is money-bank up".

Go out and buy this, you will know it!

Made with

## Tour de France

\*\*\*  
Activision  
1575  
C64 + joystick

SOFTWARE THAT GIVES A LIMITED amount of French lessons is the latest offering from Activision. Based on the Tour de France bicycle race the game boasts some interesting and well planned graphics. I was particularly impressed with the bicycle movements which allow you to pedal movement in any direction as the course demands. With each slight movement the space appears to change shape accordingly giving an extremely convincing 3D effect, probably the best I have seen.

The game is complex to load because you are given the choice of completing the full Tour or selecting individual stages which are then loaded from data sections on the tape. This means that once you have completed your chosen selection a new game involves reloading the whole program again. Not knowing how the program was written I do not understand why it is not possible to be able to reload the data section of the tape and reload a new selection. No doubt there is a reason but I would have preferred it to be otherwise.

Another little quibble relates to the fact that although the race can be controlled by the keyboard, the motion of bicycle and country of the player must be made with a joystick. Surely a small amount of work thought could have released the use from the need to use the joystick at all. I found that the response from the keyboard was better than that from the joystick and that the control of the bicycle was easier.

Having got these grapes off my chest I shall continue with my description of the game. Up to ten players can enter the race but each must complete a section in turn. Why the programmer bothered to include the option for a practice mode I do not understand because once practice

mode is chosen you must reload the tape to switch to competition mode. I found that by selecting competition mode with ten players gave me plenty of time to experiment with different types of cycle and perfect the technique of staying on the road.

To pedal your bike you must press two keys alternately or wobble the joystick back and forth. The energy expended to do this will not give you what you like some sport simulations but it does make you concentrate. Direction is controlled by two separate keys or by moving the joystick left or right with the fire button pressed and you can even select one or two gears in the bonus demands.

French is used throughout the program but most of it can be deduced without any prior knowledge of the language and who knows you may learn a thing or two.

At the beginning of each stage (stage a screen is given which allows you to select keyboard or joystick operation and it also permits you to select another race. Each bike has different qualities but it is up to the user to find out what these are.

This means that if you want to change your mode of control or try a new vehicle in real-time you can do so at the next rest point between stages.

When I first played this game I thought it was dull but when I discovered the target time for each section on the title screen for each stage, I became engrossed in pedalling like fury avoiding collisions with the grass verge so that I didn't waste time parking myself and the bike off the floor in an attempt to beat the record time. This added a distinct edge to my one player game and I can see how the multi-player game could become addictive.

At the end of the game you can load a summary screen which displays each player's overall score and gives a breakdown of individual performances. I'd like to say more but it would only be my version so I'll take Activision's word for it.

155

## Nick Faldo Plays the Open

\*\*\*  
Mead Games  
1575  
C64

AMAZE YOUR FRIENDS ON THE 18TH tee with the casually dropped remark, "I bet a 71 at the Royal is George's Club in Sandwich this morning".

Ceef arbiters and amateur critics in the should enjoy taking up the challenge of

one of the most difficult courses in the country.

Using either the keyboard or joystick you can view a detailed plan of each hole so all you need to do is select the right club (with guidance from your caddy), pick the direction in which you wish the ball to go and choose the appropriate strength for the required distance you wish the ball to travel. Then, adjust for wind direction and speed and hey presto your amateur golfer will hit the ball. It's that simple a lot too easy for you there it



## Sword of Destiny

\*\*\*\*\*  
Gremlin Graphics  
14.95  
2.75 on Plus II — Gremlin Graphics

THIS GAME WAS AN UNUSUAL STORY-line in that you start off dead! You play the part of Dethel, a not-the-composite-man-of-legend but the well-known warrior of the Isenar-Borgon. No, you don't work for Angla Television either!

You have been killed by Xophax, a warlock of the worst kind, and he has stolen your heart away. "Have a heart!" you yell, and he just took it, down to see in some of his unusual practices. So you are doomed - doomed to wander forever in the Abyss of Death, doomed to the concept of death without peace. Somewhere in the succession of some control has your lost heart, which you must regain if you are ever to find rest.

To help you in your quest, you are armed with the famous Sword of Destiny. A strange sword, this, as it kills those of people - or perhaps they are disbelievers. You use it to dispatch the evil guardians of death, an assorted bunch of bats, ghosts and gnomes and goblins, and by sending them packing you regain your energy a little. Collect any looking artifacts and you open up further sections, but do watch your energy, or actual power level. You only have one life, another death, to play with.

So each up, you must run along and jump from level to level, collecting anything which flashes and bounces off ghosts, shadows and things that go bump in the night, trying to find your poor bruised heart. The task isn't easy - it took me several attempts before I even qualified for the high-score table - but at least you get some help from the Isenar-Lum, which enable you to keep on winning help!

Here we have yet another great game from Gremlin Graphics - it's well worth buying.

P.B.B.

## Dark's Dilemma

\*\*\*\*\*  
Gremlin Graphics  
14.95  
2.75 on Plus II — Gremlin Graphics

DORIS, AS EVERYONE KNOWS, COME from Dorling, but this one is far from the heart of rural Sussex. The spaceship has crashed on the planet of the Zedoids and is scattered to the four winds, or at least to the 25 mazes. "What is a Zedoid?" I hear you ask. Elementary, my dear Watson - it is nothing more than a flakey spell backwards, but it can take many forms, all of them nasty.

If you are wondering whatever I am talking about, let me explain. The aim of the game is to collect a piece of spaceship from each of the 25 chambers, then reassemble them. To collect these components, all you need to do is to kill a minimum of 10 Zedoids in each chamber. The method of killing is easy. You drop a bomb when they are close to you, then you dash out of the way pretty sharply. The trouble with Zedoids is that they tend to surround you in a cluster, so you can't get away after dropping a bomb. You have four lives, however, so it isn't as bad as it sounds. Some of the chambers are rather cramped, making it difficult to manoeuvre, and you start in a different room each time you play, so it is hard to work out a technique for dealing with each one.

Gremlin Graphics has produced some really excellent games for the C16 and Plus II, and this is no exception. Complete with high score table, joystick or keyboard options, sound on or off, it is a thoroughly professional piece of programming. The graphics are coloured and sound, and the sound effects are just right.

The game is entertaining, and challenging enough to maintain interest. One word of advice - don't rush around too much, but be patient and you can achieve a very high score.

P.B.B.

also a "Yin-Yang" which allows you to make last minute adjustments to the strength and direction of one shot or your gutter rings his club through the air.

You can then watch the ball fly across the screen and land on the battery, or the magnet, or the bumper, or the water hazards or even go out of bounds! The usual inventory frustrations of the home-use golfer prevail. At least you have the consolation that if you miss a two-foot putt, no-one else is watching.

The weekend hackers and club hand-

cap gamers alike will relish the chance of putting their wits against an open championship course. Even those of you with no golfing experience should quickly learn enough of the rudiments of the game to play many enjoyable hours trying to fit a small white ball into a small white hole. This is a game to learn but like real golf, it's a task to master. Holes in one, eagles, birdies and pars are all attainable but, because a longer's allowed in previous supply for the winners.

P.T.



## Thing on a Spring

\*\*\*\*\*  
Gremlin Graphics  
12.95  
C16 + joystick

IT'S A LONG TIME SINCE WE HAVE seen any C16 games from the stable of such classic games as Worms, Maze and Potty Pencil. However, Gremlin Graphics has now returned with vengeance with what is sure to be its latest chart topper - Thing on a Spring.

As ever, your mission is to rid the World of some evil that has been unleashed upon it. This time, however, it's not the usual mad scientist or the Warden past that you're out to stop, it's an evil golden witch. It's trying to rid the world of all its creatures.

This time you don't play the role of a lone commander in a superhero instead you find yourself quite simply at a Thing on a Spring.

Platform games have certainly become as common as Space Invaders, and it's quite a surprise that Gremlin Graphics has refused yet another run at this format. Don't worry though, this isn't your run-of-the-mill platform game. It's far superior to most that have been stretched recently.

Controlling Thing is no easy task, being a spring he tends to bounce around quite a lot. If you're not careful you'll find him bouncing right into the clutches of some evil monster who will reduce the amount of oil on his spring. You hit a c and you guess what happens!

Hidden around the numerous rooms in the Cade is a lot of fireworks. They need to collect these before he can get around all of the components. Finding the switches and finding out what they do turn what could have been a normal platform game into a real arcade adventure. It took me an hour to get out of the first room.

Thing on a Spring has some superb graphics and sound. Gremlin's aim that Thing is 'cute and whimsical' isn't that far from the truth.

Rush out and buy it, you'll save it.

P.C.



# Software Spotlight



WHEREIMLED HAS A PASSION FOR collecting things. TV sets, grapes, toys, or, occasionally, lives. A strong pull on his acquisitive nature that he will risk life and limb to make them.

Our hero keeps in a dangerous world of rooms which are guarded by snakes and small round creatures with antennae. Whereimled must collect as many objects as possible avoiding contact with these creatures, and can fly by using a laser blade fired to his hand.

The first five rooms are fairly straightforward but the complexity increases as you move from screen to screen. Eventually, you reach rooms in which you must find keys which fit in locks of the same colour. Placing a key in a lock opens up another part of the maze which contains one of the sought after objects, but also releases another creature to add to your problems.

I found this game quite challenging to play but I can't say that it was more exciting than the host of other games in the same vein. In most of these games, once you work out a pattern to follow each maze can be traversed without a great deal of difficulty. One good feature is the ability to start at a higher level, preventing the need to start from the beginning every time.

## A View To A Kill



SOME BOMB PROGRAMMERS CAN GO overboard with special effects at the expense of what could otherwise be a terrific game. A hero in a kill is one such example.

Based closely on the plot of the film, the game is split into three parts, or less if you count the title sequence which merely compensates the familiar opening sequence to all the Bond movies.

The first scene picks up the story where Bond is chasing the hang-glider in James' May Day. He has sequenced a laser for this purpose and he must avoid collisions with the buildings and the road vehicles of Paris if he is to intercept May when the touches down. To help, you're faced with a rather complex screen which shows a 3D view from the front as well as the car, a bird's eye view of Paris and a gauge which tells you how far away May is.

Although this is an excellent piece of programming, I would prefer a larger view of Paris so that I could see exactly where the hang-glider is. Control of the car whilst leaving all the other displays is actually impossible and you can't tell if the street you are on is a cul-de-sac or if you will end up circling powerfully in circles.

The next two parts are action-adventure games. The first set in the San Francisco City Hall and you must rescue the beautiful Stacy from the life-shed where arch-villain Alex Zorn has trapped you both after setting fire to the building. As the fire engulfs the room to room you



must try to obtain the necessary equipment not only to rescue Stacy but also to use to escape from the building. Using a 'look about' menu you can examine or use the objects which you find, command Stacy to leave or wait behind and if you go disastrously wrong you can quit and start again.

The final part is set in the mine in which Zorn has placed the device which will soon explode to make sure that the occupants of St. Peter's are hurt. Before you can defuse the bomb you must search the mine for suitable gear so that May Day can help punch you down to do your work. This section is very similar to the previous one, though in my opinion it is the better of the two.

Last section of the game relies on success at two previous sections to enhance your chance of being able to complete it. For example, a greater number of bullets in the City Hall, which will help to locate the bomb in the mine. This gives a theme by which the three parts hang together as a whole but it also detracts from the fun enjoyment of the individual parts.

My advice is to spend your money on Bond but make sure that it is a Premium Bond, you have more chance of success and it may prove more rewarding.



ALTERNATIONS AREN'T AND ASSISTERS another attempt at aggregating action for US Gold. Unfortunately, I don't think that it will, not because it is a bad game but because it offers nothing more than the repetition of battles and platform games currently available.

After a rather crude US Gold standard loading sequence the title screen gives way to a view outside the library where we find out post-be-and

hard standing outside three doors. The doors are marked Control, Protection and Shipping but as the game is unimpressive matter which door you enter we don't agree on the significance of these labels.

On entering one of the rooms you are faced with a scene of part of the burning factory. There are doors on most of the level and one or two of the floors have railings. Somewhere on the screen is a key which unlocks the way to another room and it is then's aim to push the key and move on.

Next, a door is closed an underground of Bert jumps on one of these he is carried by the staff up as to the next floor. As escape from some of the rooms the air lock provides the only pathway to the next.

Occasionally one of the doors opens

and a mass like character appears. He walks up and down like a security guard and will fire the odd pot shot at Bert who must dodge or die. Bert is also armed and he can retaliate with a well-aimed shot which will kill the opponent who then disappears only to re-appear from behind another door.

Some of the rooms are linked and the only possible means of escape is to enter the other rooms and face its peril only to reappear on the level itself of the original room to make his escape.

For my money, this is not as good as the Bounty Bots game which US Gold markets but as you go from room to room the challenges increase in number and variety and I found that it at least held me interested.



This month RuneCaster delves  
into the secrets of Exodus:  
Ultima III and encounters  
elves, dwarves bobbits and  
tuzzeit!

## It's time, uh, time!

ARE YOU A FRUSTRATED BARBARIAN with your sword rusting on the wall? Perhaps, if you're having trouble resting between quests. Did you want to play Dungeons and Dragons? You couldn't find anybody else to join you at the right time or place? Well, for the hours (days?) necessary for a worthwhile quest...

Get down that sword, oil the rusty joints, get in a goodly amount of exercise and prepare for an extended period away from the daily grind.

If you haven't got a C64 complete with disk drive, now is the time to invest... and, while you're at it, get a copy of Exodus (Ultima III) by Sir-Gilbert Siersma Inc., distributed in the US by US Gold. This is such an interesting find that for the first time I shall be looking at only this one subject. I hope you like role playing games.

Initially, pleasant, but hard-earned, all been for single adventures (no training or level-up usually) had that nearly had alternative reality (that makes for total player involvement).

### Presently Available

Other computers, especially in Australia, have had good role playing games written for them. Naturally for the Apple it was that immediately spotting to mind. Now, at long last, C64 users in the UK have their...

complete you have to switch off the computer (and disk drive) and reboot the operating system before you can play. The sole difference in "playing" and "resting" is whether you press the SPACE bar or "N", since the main menu appears. Don't make a mistake, as even loading the operating system takes several minutes.

This isn't a game to be played lightly. It takes time to set up and even more time to get over the initial learning period. What time do other, which is the best way to make up your mind of adventures (you can control up to four in a party).

For this reason I could suggest a letter



### in the Box

Some four years ago there was the Valley, where you could choose your character type (fighter, cleric, wizard, etc.) and so on, adventuring with battling a host of monsters, finding treasure and casting the odd spell. At the time it was quite a hit but with limited graphics and designed for the MS-DOS, it now looks a little patchy.

There have been several attempts to bring genuine turn-based role playing games to the C64, notably perhaps the Supersword series which included The Inside of Aches. They have been

... (I) could well prove to be a game which adventures will spend more thousands of hours playing. Sadly, it's still based only with manual hope of a cassette version ever appearing.

The "operating" system for the adventure comes on one side of the disk, with a program for creating a separate scenario disk on the other side. Before you can venture forth you must create this separate scenario disk.

This creation routine takes some time (about 15 minutes). When

reaching to start with time through you will be trying to get going. Make a scenario disk! Make sure you know the data with exactly the same disk name and ID.

We also now have a way to start using the operating system. Up comes the main menu, press the SPACE bar. Within, when not long to wait long, insert your scenario disk, when told to, and recharged.

The initial screen displays a moving picture of a team travelling around in boats fighting monsters and generally



enjoying themselves. You could search this for longer, but what about your party now?

In *Process*, *SPACE* gives you three options: Return to the meeting room, organize a party, journey toward. You cannot choose the third option (the creating your characters and forming a party are your initial tasks) if determined for you.

Again a menu, giving the option to examine the "main" characteristics of existing characters, creating new characters, forming or organizing a party, examining *Characteristics* information and finally returning to the previous screen.

## Vital Statistics

Initially you must create a character or characters (at least four). You can go adventuring with less than four but this is not recommended, especially in *Dark* (1).

You may have up to 10 characters "in play" and in creating each one, you must select their race number (1-25), sex, age, size (height), all should be built in *Dark*—each would have their different personal statistics. There are strong, slow, clever, dexterous etc. You must have this in mind—your results must reflect what would you.

When you have all created the characters, you'll be back to the main screen etc. There are 15 positions (you and the others must be made carefully).

Most early do you want a balanced party that can hold its own? It seems to be the way but you must also attempt to put in a total of 10 to your character's strength, dexterity, intelligence and wisdom. They together with race, sex, age, size, will really a character's capabilities.

The permutations possible are quite enormous but after a few hours into the unknown you will start to get some idea of a viable party's requirements. You will probably "come to" as a trifle of some sort a probably one that can also use a little magic.

Then you'll need someone who can use weapons (they should not). It isn't because there can be magical items and also magical weapons (weapons should be there). I haven't seen so far very useful a powerful 10 in the battle, character's. Then perhaps the pure blood type or some pure magical and magical but not so.

Finally, the table, after you, the character's for each position. It will tell you what weapons, armor and spell (magic) each can use, what other companions it needs (if any) for the use of magic.

Make notes of each character's attributes as you type them in. This is a *Characteristics* because once created, you cannot change them. You cannot check each individual's position in strength, dexterity, intelligence and wisdom.

You may always call up a character's stats during play but once the adventure proper has started you cannot return to the "main" menu. When you have created sufficient characters (this "form a party" meaning the four characters you have chosen, then return to the main menu and journey toward).

## Cheat

If you wish to cheat a little, press "C" which allows you to save your position, remove the scenario disk and watch everything still. Here, reload the program opening system but this time put in the second scenario disk you created.

Go through the character creation and party formation routines, giving exactly the same information as you did the first time. The scenario disk records all your characters' changing stats as the game progresses, so loading their stats.

As this is quite likely to happen later, save to start with one of your "cheat" data should be kept as "clear" or possible (as this and when you have each one and "save your position" to the "clear" one as your character's gradually improve their standing).

It is almost impossible to get into combat without knowing about their active position, but at least it will show you a few details of their being and position. Character's who you would rather be learning how to play.

## At Large in Sounia

You are there to live, with the world in your hands, standing with open rolling plains behind you and a small walled city in front.



of you. The display shows a grid view, with one character indicating the position of your party, some it seems to be the right of the map and some buildings represent a castle and the small town. You can see some woods to the west.

To the right of the map is a "list" display showing all your four characters. It's good you go to a screen, with information on your own character's stats, position, (strength), level, experience, etc. (using player's) and their level (all start at Level 1).

Further stats (intelligence, dex, int, etc.)

By typing "C" (cheat), you will tell you how much gold they have, what weapons or armors they are carrying and their current strength, dexterity etc.

## Naked into the World...

Each character starts with a dagger and cloth armor (1) but you must tell them to "look" the dagger for use and "wear" the cloth armor—otherwise they will soon be lying along their bare backs and to their death.

There are also provided with 200 pieces of gold. The trouble thing to do next is "look" the town and view, view the weapons shop or the treasury. Weapons may be bought and sold at the local price (but haggling) as also may be food when you need it, from the grocer.

## Just a Quick One...

There is also a job where local *Dark* may be found (thought). In some areas you may also find some useful places, but you always need to search for these only slowly. You can even talk to the local people you meet in the towns.

You are now allowed to be the best of your power and have knowledge of the town. You, here of, *Dark* read the instructions that come with this game or instructions that have already been attached all they tell things and you perhaps had noticed just some of the confusion in the instructions is a lot of it. It's a little bit only one is explained.

The spells have four levels (magic) and are: *REPAIR*, *MAGIC*, *POISON*, *SANCT*, *DRAGON* (each level of these depends upon the number of magic).

When a character has them, he has depend upon character type and how you allocated them original 20 points. Only by playing, can you begin to understand how to achieve a workable balance.

## Spell That Sounia

Use the spells as needed (some), while watching along in battle, but understand, etc. You if you can work out what they do, it doesn't hurt. You only begin to play for about 20 hours so *REPAIR* and







Barry Miles has been looking at Handic's box of tricks and he found one or two surprises.

THIS PRODUCT IS A VERY INTERESTING combination for the C64 user. It is both a motherboard to enable up to three cartridges to be installed simultaneously and also an IEEE interface.

If you own an earlier Commodore machine you may already own a disk drive. The early models used the IEEE bus parallel bus, normally known as the "1-megabit" bus. This was a slight variation on the original design by Hewlett Packard.

When the VIC 20 arrived, Commodore, to many people's amazement, changed to a serial system. This resulted in data being piped along the line with the bits following one another rather than eight bits going in parallel. Obviously, this slowed down the data transfer rate and the 1540 disk drive used on the VIC 20 and the 64 has been slated for its slow performance ever since. This same problem has been adopted on the C16 and the Plus4.

The Plus4 Commodore user may buy some form of interface to make the 4040, 8050, 8250, 1081, and 2031 drives work with the 64. Certain Commodore systems interlock - happily still available - solves this problem but enables these disks to communicate with the 64 at only the same speed as the VIC.

What the average user wants is to be able to take advantage of the maximum speed of which the disk drive is capable.

Generally speaking the interfaces which are available rely on placing some machine code in the memory of the 64. This sometimes conflicts with memory demands made by advanced packages.

There are difficulties when it comes to connecting a number of cartridges to the 64 and switching between them. Many and more ingenious manufacturers are putting protection into their cartridges which causes problems.

Handic's approach was to supply an IEEE interface board enabling the user to choose between three cartridges plugged in at once, while at the same time being able to use Commodore disk drives at full speed. In addition, the product has a Reset button.

This was an interesting aspect of the design because it combined three desirable features. It is frustrating to plug in another board in order to use your cartridges only to find that you cannot use your disk drives.

The Superbox is a handsome product and looks very durable. The rigid metal case looks as if you could drive a car over it without damage.

# SUPER BOX 64

The only protrusions are the edge connectors. One for your IEEE cable and the other for plugging into the cartridge slot on the machine. There are rubber feet on the bottom to make sure that the box will sit flat, two keys on the 64 are used and this avoids any memory conflict.

In order to pick up the necessary signals it is necessary for you to connect a crossover cable up to a single monitor on the C64 board. This is not a difficult task to do but service and home-brew users may prefer to get their dealer to do this for them. Fortunately you can remove the box, should you need to do so, while retaining the connection.

The switching has been very pleasantly designed indeed. At the far end of the box, is a set of five switches. As you switch on one cartridge slot the other switches pop-up - the only way you can accidentally have two cartridges in the chosen slot once is by pushing two buttons simultaneously.

The IEEE busmen will connect you to your parallel IEEE device. The Reset button is a great advantage because the 64 unlike its successors, the C16 and the Plus4, had no reset button whatsoever. By pressing the red reset switch a "Cold start" is performed which leaves the contents in RAM unaffected. If the system has "hung" one of the buttons may get you out of trouble without losing any data.

## In Use

Feeding the box with a variety of cartridges showed that most of them would work perfectly OK. Certain types of cartridge however did not function correctly.

There is a somewhat startling warning on the front of the box manual saying that if an IEEE unit is not connected to the board and not patched on when the machine is

running, the Superbox 64 may be damaged. However, Handic L.B. confirmed that this was merely an example of Swedish caution and not to be taken too seriously.

You are warned not to connect more than two IEEE units to the computer and to make sure that all units connected are switched on before running the system. I was not tempted to test this out with a variety of extra IEEE units.

## Conclusion

At £85, the Superbox 64 is not a cheap product in fact if you compare it directly with another mother board you will find it is extremely high priced. However you are really getting two products for the price of one I highly recommend.

The only difficulty likely to be encountered in using the Superbox arises because of its all-in-one working approach to life. You are expected to be using either all Serial or all Fast all IEEE devices. An addendum to the manual gives a guide to this.

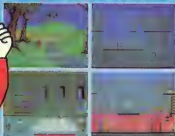
By plugging fast memory locations and recording their content manually, you enable the use of the Reset button without the loss of a Basic program. Thus, you press the button to switch between IEEE Serial and Parallel in either direction and then press the Reset button. Finally, you POKE the data which you have recorded into the same memory location.





# RUPERT

TOYMAKER'S PARTY



Jolly old Rupert can walk, jump, leap, climb stairs – and even fly – but can you get him to the party?

32 challenging screens of animation and fun.

## QUICKSILVA

The most powerful games in the Universe

Quicksilver Ltd, Liberty House,  
222 Regent Street, London W1R 7DB  
Telephone 01-439 0666

Rupert Bear © Express Newspapers PLC

Spectrum 48K  
CBM64

£7.99



# OR... MADE BY ALIEN



## THE FINAL CARTRIDGE

- A new outside OPERATING SYSTEM built in a cartridge, that does not use any memory and is always there, compatible with 99.9% of the 20,000 C64 programs.
- 24K EXTRA RAM for basic programs.
- A DISK TURBO that can only LOADS 4 TIMES FASTER than disk and SAVES 4 TIMES FASTER.
- A PRINTER INTERFACE that is compatible with all the well-known electronic printers and Commodore printer programs. Prints out all the Commodore graphic characters and control codes.
- A SCREEN DUMP for normal and three screens (1 to 3). Doodol! With a special built in routine that saves for the memory address of the three pictures automatically.

FOR C64-66



£50.-

- PREPROGRAMMED FUNCTION KEYS: Run, Inc, load, save, catalog etc.
- MACHINE LANGUAGE MONITOR: Very lightweight assembler/disassembler/monitor.
- BASIC 4.0 and DOS COMMANDS: Dload, Dsave, Catalog, search directory without disturbing your programs.
- BASIC TOOLKIT, etc.

Wanted! (and good) help in RESULT SWITCH. Resets for unwanted. Resets with old format your basic program instantly. Resets every protected program.

• CASSETTE TURBO that LOADS and SAVES programs 50 TIMES FASTER including files and records.

**H.P.**  
H. P. H. P. H. P.

Wolphaertshoofd 236  
1083 PW Rotterdam,  
Netherlands  
Tel: 01031-10231982  
Telex 26401 a hrs nl



## A.P. and D.J. Stephenson explain the use of code other than decimal and hexadecimal.

PART 1 OF THIS MONTH DEALT ONLY with decimal and hexadecimal codes because they are the most popular for general purpose use. However, there is another method of representing binary known as Binary Coded Decimal, (BCD for short). The 800 microprocessor is particularly well equipped for handling this type of code. The common arithmetic primitives of add, subtract and multiply are, by default, carried out in normal two's complement binary. It is possible to change this by using the instruction `BCD` which is the mnemonic for 'BCD Decimal mode'. Once the machine has executed `BCD`, its code will keep all subsequent arithmetic in carried out in BCD until cancelled by `CLD` (opcode 109) which is the mnemonic for 'Clear Decimal mode'.

### The BCD code

Before considering the areas where BCD might be useful, we must understand the mechanism of the underlying code. Perhaps the easiest way to start is by examining the following byte, `AB041111`,

# MASTERING MACHINE CODE

split into two separate nibbles for ease of understanding.

1001 0010

If this was interpreted in terms of a (decimal) unsigned binary, it would represent 146 in decimal, 2164128. If interpreted as signed binary (two's complement) it would represent -113 in decimal. Finally, if interpreted as BCD, it would represent the decimal number 92. In fact, the space we left between the two nibbles, although left blank, turns out to be a direct clue to the

understanding of the machine's binary BCD. Each nibble should be thought of as a separate block box, the right hand box represents the unit's column and the left hand box represents the tens column. Here are some BCD examples:

1000 0111 = 87

0110 0001 = 61

0110 0011 = 77

0101 0000 = 10

### Redundancies

Apart from the difference in interpretation, the BCD code has, what the code writers are pleased to call 'redundancies'. We can see what this means when we consider the number of possible ways in which 8 binary bits can be arranged. For simplicity, let's start with two bits. These can only be arranged in four ways, 00,01,10 and 11. These four can be arranged in 8 ways, 000,001,010,011,100,101,110 and 111. So for two bits, there are four ways (2<sup>2</sup>) and for three bits there are eight ways (2<sup>3</sup>) so we can deduce from this that the general formula for knowing the number of ways of arranging *N* bits is 2<sup>*N*</sup>. Here, in the case of an eight bit byte, we shall expect that there are 2<sup>8</sup> = 256 ways of arranging the bits. Normal binary and hexadecimal codes use a 1:1 these combinations, but BCD doesn't. In fact, as the following table shows, there are six legal combinations (redundancies) in each nibble.

BCD	Decimal	
0000	0	1000
0001	1	1001
0010	2	1100
0011	3	1101
0100	4	1110
0101	5	1111
0110	6	
0111	7	
1000	8	
1001	9	

illegal combinations

Program 10.1, Convert BCD digit to ASCII

```
10 033C      ! CONVERT ASCII TO BCD DIGIT
20 0000      #=BCD000
30 0000      MEM      =  #FB
40 0000 3B      BDC
50 0001 A5FB      LDA MEM
60 0003 E730      BDC #430
70 0005 A5FB      STA MEM
80 0007 40      RTS
```

Program 10.2, Converting ASCII to BCD digit

```
10 033C      ! CONVERT BCD DIGIT TO ASCII
20 0000      #=BCD000
30 0000      MEM      =  #FB
40 0000 1B      CLC
50 0001 A5FB      LDA MEM
60 0003 A730      ADC #430
70 0005 A5FB      STA MEM
80 0007 40      RTS
```



As you can see, the illegal combinations are those greater than 1001. These are the combinations normally represented by the letters A to F in hexadecimal.

We conclude, therefore, that representing numbers in BCD is neither easy because, out of 16 possible combinations, only 10 are used. In percentage terms, this represents an efficiency of only  $(10/16) \times 100 = 62.5\%$ . As far as memory usage is concerned, we need about 30% extra memory space to store numbers of BCD form (roughly, relative to all required for each BCD digit in spite of the redundant digits which remain unused). We can represent this only in a more obvious way by comparing BCD with unsigned binary in respect of the largest number capability. The largest legal combination in BCD is 1001 1001, 99 in decimal, whereas in unsigned binary the largest is 1111 decimal.

## BCD addition

Since we mustn't use combinations greater than 1001, what exactly happens when we add 1 to the BCD number normal binary conditions. The addition would be quite straightforward:

```

0000 1000
+0000 0001
-----
0000 1001

```

But, the right-hand mode now has an illegal combination! However, if we add had been only binary not strictly by means of BCDs to perform the addition in BCD, it would perform as follows:

```

0000 1000
+0000 0001
-----
0001 0000

```

Note that adding the 1 has now led to a permanent carry from the right-hand nibble into the left-hand nibble. It is as if the 'half carry' (or 'daught' as it is called) from the normal carry out from the right-hand nibble is added to the next 16 in decimal is correct in BCD format. How does the microprocessor perform this bit of 'magic'? Actually, the processor simply performs the addition in normal binary and then uses the result of the right-hand nibble as 'legal' if there's a further carry. This may seem superfluous but is quite logical when you think about it because there are no illegal combinations which need be checked. Examine the following example, using the previous figures:

```

0000 1001
+0000 0001
-----
0000 1010      illegal, so

+1 = 0000 0110
-----
0001 0000      result correct in BCD

```

## Program 13.3 Convert hex digit to ASCII

```

10 033C      ! CONVERT HEX DIGIT TO ASCII
20 0000      #=BC000
30 0000      MEM
40 0000 10      = #FB
50 0001 00FB    CLC
60 0003 090A    LDA MEM
70 0005 9002    CMP #10
80 0007 6906    BCC OVER
90 0009 4930    ADC #40
100 000B 05FB   STA MEM
110 000D 60      RTS

```

## Program 13.4 Convert ASCII to hex digit

```

10 033C      ! CONVERT ASCII TO HEX DIGIT
20 0000      #=BC000
30 0000      MEM
40 0000 30      = #FB
50 0001 00FB    SEC
60 0003 0730    LDA MEM
70 0005 090A    SBC #30
80 0007 9003    CMP #10
90 0009 0907    BCC SKIP
100 000B 05FB   SKIP STA MEM
110 000D 60      RTS

```

You should try out the code with various combinations to convince yourself that it works every time.

## What use is BCD?

Since BCD has been demonstrated to be inefficient, the obvious question is what do we get in return? One answer, not the most important, is simplicity. Only a cursory glance at the contents of a BCD byte is sufficient for most all to be converted into the equivalent decimal. However, a more substantial reason for including BCD in a computer is to make it compatible with interval equipment, particularly the vast army of digital-controlled instruments. Most instruments of this kind are designed to accept BCD formatted inputs or deliver BCD outputs. For example, a digital voltmeter can deliver readings from an external system to the computer in BCD 'language'. Conversely, a variable voltage power supply can be controlled by

feeding BCD information from the computer.

It is obvious, of course, that the majority of modern microcomputers which operate in which case these facilities may be of little interest. But, we should remember that any computer has vast potential. It is possible that sometime in the future, maintaining though they may be for a time, may not last for ever and many users may feel the urge to expand their machines in other directions. We should bear in mind that the input/output bus in Commodore machines, even since the days of the original PET 2001, have been based on an industry accepted standard known as the IEEE bus protocol. Thus, the Commodore microchip is a fully modulated version of the bus (i.e., in general, it is reasonably compatible). We have no space in this article to discuss the details of the bus although we should point out that BCD is the accepted code used for passing numerical data between computer and external equipment.

BCD is also useful as a convenient



Program 12.1 Add two BCD numbers and display result

```

10 033C      1ADD TWO BCD NUMBERS AND DISPLAY
20 033C      THE RESULT IN DEC DIGITS
30 033C      NUMBER1      = #B8
40 033C      NUMBER2      = #9C
50 033C      RESULT       = #FD
60 033C      CHROUT       = #FFD2
70 0000      **BCD000
80 0000 FB          DEC
90 0001 A908        LDA #8
100 0003 89FB       STA NUMBER1
110 0005 A909        LDA #9
120 0007 89FC       STA NUMBER2
130 0009 18         CLC
140 000A A5FB       LDA NUMBER1
150 000C A5FC       ADC NUMBER2
160 000E 89FD       STA RESULT
170 0010 201800     JSR OUTPUT
180 0013 08         CLD
190 0014 60         RTS
200 0015          I
210 0015 40         OUTPUT PHH
220 0016 4A         LSR A
230 0017 4A         LSR A
240 0018 4A         LSR A
250 0019 4A         LSR A
260 001A 18         CLC
270 001B A930       ADC #30
280 001D 2002FF     JSR CHROUT
290 0020 48         PLA
300 0021 390F       AND #0F
310 0023 A930       ADC #30
320 0025 2002FF     JSR CHROUT
330 0028 60         RTS

```

intermediate code useful as a stepping stone for other conversions.

### Code conversions

The above discussion on BCD raises the general question of conversion between codes. It is often required to convert information from one code to another, either for reasons of efficiency or convenience. The conversion can be achieved by what program segments primarily written in machine code. For the most part, they are short so instead of the usual practice of preserving an additional hex dump, we shall present them in full assembly format which includes the equivalent op-codes in hex.

### Convert BCD digit to ASCII (See Program 12.1)

Program 12.1 Convert BCD digit to ASCII

For the benefit of those who do not have an assembler, the columns, reading from left to right, have the following significance:

Column 1 is a line number, used for reference purposes only. (You can't jump to a line number here we do in BASIC.) Column 2 is the hex address of the location which stores the first byte of the current instruction.

Column 3 is the instruction in terms of machine code hex digits.

Depending on the instruction, there may be one, two or three pairs of hex digits. The first pair will always be the op-code and the remaining pair, if any, will be a single or double byte operand.

Column 4 is the label field. That is, entry, it will contain assembly chosen labels for branch destinations. If no labels are used the column is left blank.

Column 5 is the instruction in assembly language, consisting of the mnemonic

code and operand.

Those without an assembler will have to place Column 3 machine code bytes in the data statements of the hex loading program given in part 3 of this issue (Printer's note) Taking program 12.1 as an example, the data statements would be entered as follows:

DATA 033C,033C,033C,033C

### Description of program 12.1

Line 10 starts with 1 so is simply a comment. Line 20 sets the accumulator to locate the program starting at the address 0300. (In our numbering, 3 means hex, not string.) Line 30 tells the assembler that the address 800 is to be known by the symbolic name NUMBER1. It is assumed that the current contents of 8004 is holding the BCD digit to be converted.

Line 40 clears the carry ready for the later add.

Line 50 loads the accumulator with the BCD digit.

Line 60 uses indirect addressing to add hex 30 (decimal 48) to the BCD digit (space 0000000000000000). When 48 is added, it would find 01 which is the ASCII code for 1.

Line 70 stores the converted number back in the original location. The program can be used as a sub-routine called frequently in a BASIC program by using 205-49151.

This simple program has been created in extra detail because the full assembly format may be new to some readers. It is hoped that the programs in this follow can be understood without resorting to a similar line by line analysis.

### Converting ASCII to BCD digit (See Programs 12.2)

Program 12.2 Converting ASCII to BCD digit

This is actually the mirror image of the previous program, the only difference is that 30 is subtracted instead of being added. However, we should note that the carry has to be set, using BCC, before a new subtraction instead of being cleared by CLC.

### Convert hex digit to ASCII (See Program 12.3)

Program 12.3 Convert hex digit to ASCII

Line 60 contains the value of the hex digit in BCD form. If the carry is more than 70, it might be that it was more than 99 so a branch is made to line 0000000000000000 to put 0 into the ASCII field.

If however, the previous comparison



showed that the hex digit was 75 or greater, an extra one is added to it one for the gap in seven between the ASCII code for 9 and the ASCII code for A, hence that the carry is always set before the ADC. It is irrelevant, so, in effect, we are adding 7

## Converting ASCII to hex digit (See Program 12.4)

**Program 12.4** Convert ASCII to the hex digit

The program is almost a mirror image of the previous one, except of course that subtraction has been used instead of addition which is why line 40 sets the carry. Hence that, this time, we add 7 is subtracted (the gap between ASCII 7 and ASCII 'A') since the SBC instruction is entered with the carry set.

## Addition of BCD numbers (See Program 12.5)

**Program 12.5** Add two BCD numbers and display result

The program is intended only as a guide to the general procedure of adding in BCD because of this, the sample constants used have been used for the two numbers. The result, 17, is arranged to be printed out on the screen using the ROM subroutine CHROUT (the user must remember that CHROUT will only pass the character corresponding to the ASCII code in the accumulator.) Practical versions will, of course, be more likely to supply the numbers in variables.

## How the program works

Lines 10 to 40 assign the symbolic address, HL40001, HL40002 and HL40003 together with the ROM character output subroutines at address 1100 hex line 40, by use of the instruction 1ED, moves one of the 1602's ability to perform all following arithmetic in BCD. Lines 50 and 100 store the arbitrary constants for later use. Lines 110 to 140 performs the addition of BCD, the add-6 operation is carried out automatically by 1ED. Lines 170 and 180 store the result and then call on OUTPUT. The machine is then reset for normal binary instructions by use of CLD. The subroutine OUTPUT is used to prepare the accumulator for action by CHROUT.

A copy of the use data sheet describing the stack in use at PHA. The accumulator at this point now contains

```
0002 0011 (17 DEC)
```

The accumulator is then printed right 4

**Program 12.4** Displaying contents of location in hex...

```
20 033C      10DISPLAYING THE CONTENTS OF A
30 033C      11LOCATION IN HEX DIGITS
40 033C      NUMBER      = 00FB
50 033C      CHROUT      = 00FD2
60 0000      00=0C000
70 0000      !
80 0000 00FB      LDA NUMBER
90 0002 200400      JSR OUTPUT
100 0005 40      RTS
110 0004      !
120 0004 40      OUTPUT
130 0007 4A      PHA
140 0008 4A      LSR A
150 0009 4A      LSR A
160 000A 4A      LSR A
170 000B C90A      CPH #10
180 000C 9002      BCC OVER
190 000F 4904      ADC #4
200 0011 4930      ADC #30
210 0013 2002FF      JSR CHROUT
220 0014 40      PLA
230 0017 290F      AND #00F
240 0019 C90A      CPH #10
250 001B 9002      BCC OVER2
260 001D 4904      ADC #4
270 001F 4930      ADC #30
280 0021 2002FF      JSR CHROUT
290 0024 40      RTS
```

places in order to produce the higher order digit at the right

```
0002 0001
```

The addition of 30 hex is then added to let the number into the ASCII based, the accumulator will now contain

```
0011 0001 (17 hex, 49 decimal)
```

This is the ASCII for the digit '7' so when the subroutine CHROUT is called, the result appear as the first digit result on the screen. The original one digit is pulled back from the stack using PLA so the accumulator now holds, as before

```
0001 0111 (17 DEC)
```

We now AND with:

```
0008 0111 (OF hex)
```

which produces

```
0008 0111
```

At 000C 000 000, the accumulator now holds only the least significant digit (7) of

the original BCD result. The result 30 hex is then added to convert to ASCII so the accumulator now holds 37 hex (55 decimal) which is the ASCII code for 7. The accumulator is then sent to the screen via another call to CHROUT for displaying the second digit of the result

## Displaying contents of location in hex (See Program 12.5)

**Program 12.5** Displaying contents of location in hex

This program does not purely display explanation since it is closely similar to the previous one. This time, 510 is not used, we expect that somewhere along the line, we have to add six, use line 120

## Summary

This article has attempted to show some of the various techniques which can be used for code conversion. Consequently, the programs should be interpreted only as guide lines to be introduced in practical programmes







# ALL MIXED UP

**This article from Gareth Thomas gives you an insight into mixing machine code and Basic.**

IF YOU WRITE MACHINE CODE JUST to improve your Basic programs, there comes a time when you BASIC routines cannot stand alone and need values passed to them from Basic. The simplest way to achieve this is by FORing the relevant values into memory and passing them out using your routine. If you do this, the best locations to use are those used to store the 800's registers which are accessed during a SYS call; they are as

- 780 Accumulator
- 781 X register
- 782 Y register
- 783 Status register

Perhaps the best known example of this method is a Basic program to PRINT AT using the CLEAR routine PLOT:

```
10 POK 781,ROM
20 POK 782,COLUMN
30 POK 783,0
40 SYS 65530,8154 call PLOT routine
50 PRINT "Pretty cumbersome though isn't it?"
```

Unfortunately, this method only allows you to use integers from 0-255.

If you need to use a bigger number a better method is to use the Basic L&R functions (I've never seen this used in a program which is probably due to the lack of documentation about it in the manuals). The L&R functions exist in the passing of one floating point number to the range 0-65535 in your BASIC program. The function takes the value converts it to floating point and places it in the floating point accumulator 1 and then returns the VMC routine pointed to by the low/high bytes

of 785/786. All your BASIC routine needs is to use the routine CCPRM at 8017 which converts a floating point number to FPACC1 to an integer in 814 and 815.

```
10 ROM set USRADD to $CAA4
20 POK 781,70
30 POK 782,30
40 P-LSR,814,815 ROM evaluates 65535 to FPACC1 then instructs routine at $CAA4
```

VMC routine	814	8017	'convert FPACC1 to integer in 814 and 815'
CAAR	104	814	'save low byte in X-reg'
	815		
	104	815	
CAAR	815		'save high byte in Y-reg'

The third method is slightly more complicated but is definitely the best, that is, extending the SYS call to pass values as well as SYS add, value 1, value 2, etc. This can only be done at machine level and needs the use of a number of ROM routines.

The first of these is CHARGOT. Although this resides in ROM, a copy of it can be found located in 7180-PAGE at location 8073. It is used by Basic to collect bytes of the current instruction (and is therefore in one-page for speed), in fact we only need a part of it called CHARGOT at 8079. This collects the current byte without first updating the pointer, otherwise the point at 8079 would be updated and we would lose the last parameter. The second routine is CHRCOM at 845D. This checks for a comma after the address, if it does not find one a "NTAX ERROR" is generated. The third routine is FVAL at 84C6A this is a complex routine which evaluates an expression and converts it to floating point and then places it in FPACC1. The last routine GINT at 8477 I have already described.

To illustrate how to use these routines I've written a small program which is a BASIC version of using 1 to use the routine

set position for PRINT but without all the FORs, just one SYS call

80	8079	'get byte'
80	845D	'if not a comma generate error message'
80	84C6A	'evaluate expression'
80	8477	'convert FPACC1 to integer in 814 and 815'
104	814	'load low byte of value 40'
800	815	'no to get next parameter'
800	8477	'no to generate ERROR'
		CLAREITY
		'save value on stack'
NEXT	814	
80	847D	
80	84C6A	
80	8477	'repeat as above for second parameter'
104	814	'> than 25'
800	815	'no to generate error message'
800	84C6A	'no to transfer to X-register'
800		'retrieve last value'
800		'transfer to Y-register'
800		'clear carry bit'
800		PLOT will only need cursor position'
80	8110	'increase PLOT routine'
815		'return to BASIC'

As you can see, the process is repeated for each parameter except that CHARGOT is only used once. Also, the advantage of using FVAL at 84C6A, is that expressions such as 3\*POC(5) or variables can be used. Really, to use the routine

SYS add,value,low PPRINT"message"





NICK FALDO  
PLAYS THE

# Open

£9.99

Runs on both  
Spectrum 48K  
and CBM64

Average  
900 screen  
scrolling map of the  
Royal St. George's  
Golf Course, venue for the  
Open Championship 1985.

Can you make par on this  
course of Champions? With strong  
winds and deep bunkers you'll need all  
the help your caddy can give you. Full  
animation and live action bring your game to life.

MIND  
GAMES



Open Press Software Ltd., 1, Derby House,  
31 Regent Street,  
London W1K 1TQ.  
Telephone 01-439 0000.

Free booklet featuring maps and  
history of the course with a hole by hole  
guide to  
*Nick Faldo*







## In operation

As supplied the *Amibios* will function with virtually all printers. To get the best out of your system, you set up the three dip switches to match the printer type which you are using (used an Epson 1380). The Universal setting will enable you to use a *Sharpwell* printer, although of course the graphics capability is then lost.

Switch 1 controls the ASCII character reverse, Switch 2 selects the device number. If you change the switch settings after power-up you must press the Reset button.

## Commodore 1525 emulation

When *elapitch* number 5 is in the off position, you obtain an exact emulation of the 1525's performance. If you send the appropriate control codes you will be able to do double-width printing, or have columnar tabs up to 60 print positions available by sending *CHGR* followed by 2 numbers which specify the start column. You can also set up your own user-definable character set and even do dot calculation, taking your print head to any one of 480 dot positions. There is a *Graph* Repeat function which repeats a single graphical character.

In the *Imulate* mode, you are also able to use additional commands which are peculiar to the *Amibios*. By opening a file to your printer with a secondary address

of 1, you arrange that all the commands which normally result in a reverse field character appearing as a BASIC program is listed, and are now translated into readable *elapitch* commands. This makes your listings is unambiguously obvious, even if you have not memorised the Commodore symbols.

In the mode you are not able to print Commodore graphs as graphics. What is printed instead is the key which would be struck on the keyboard in order to produce that graphical symbol. This is particularly helpful for printing our programs which use vertical or horizontal bar graphics symbols, which are very difficult to design as they are another.

You can also set the left and right hand printer margins to leave plenty of space for notes or to perform the *elapitch* for listing purposes.

## Sending control codes

The *Transparent* mode of operation enables you to send codes to your printer which control various special features. You merely open the file to the printer with a secondary address of five. Once this is done all print commands are sent literally. Problems could arise if you wanted to send commands to the printer while you were already in *Imitate* mode. The Commodore 1525 itself reacts to certain escape sequences. However this has been catered for by arranging that the

control sequence will be correctly translated to your printer if you precede it by an escape (ASCII) character.

The *Amibios* has a number of extra features to make life easy. You will find a world to keep the interface close at hand, because the switches and buttons are likely to be in continuous use. The push buttons between the interface. If you are using single sheet mode, the printer pauses at the bottom of each page. You simply press the button once to resume printing. There is a monitor which is used to check whether the interface is working satisfactorily. You can defeat the right margin of 60 columns if your printer has a wide platen and you need to use that.

There is a 'disappearing performers' feature, which is disabled on power-up. This is combined with a form length setting command which enables you to accommodate different sized sheets of paper.

## Conclusions

The interface has a wide range of capabilities, is very easy to use and fits a substantial need. Whilst quite expensive, its versatility, in offering Commodore users a wider selection of printers, justifies the price. And, the *Imitate* mode prevents the frustration of not being able to use Commodore printer-emulating software. The *Amibios* is well designed, robust and is highly recommended.

1. CHECK GRAPHIC				2. check business			
(COURSE UP)		MODE		(COURSE DOWN)		MODE	
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
22222222	22222222	22222222	22222222	22222222	22222222	22222222	22222222
33333333	33333333	33333333	33333333	33333333	33333333	33333333	33333333
44444444	44444444	44444444	44444444	44444444	44444444	44444444	44444444
55555555	55555555	55555555	55555555	55555555	55555555	55555555	55555555
66666666	66666666	66666666	66666666	66666666	66666666	66666666	66666666
77777777	77777777	77777777	77777777	77777777	77777777	77777777	77777777
88888888	88888888	88888888	88888888	88888888	88888888	88888888	88888888
99999999	99999999	99999999	99999999	99999999	99999999	99999999	99999999
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
22222222	22222222	22222222	22222222	22222222	22222222	22222222	22222222
33333333	33333333	33333333	33333333	33333333	33333333	33333333	33333333
44444444	44444444	44444444	44444444	44444444	44444444	44444444	44444444
55555555	55555555	55555555	55555555	55555555	55555555	55555555	55555555
66666666	66666666	66666666	66666666	66666666	66666666	66666666	66666666
77777777	77777777	77777777	77777777	77777777	77777777	77777777	77777777
88888888	88888888	88888888	88888888	88888888	88888888	88888888	88888888
99999999	99999999	99999999	99999999	99999999	99999999	99999999	99999999
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
22222222	22222222	22222222	22222222	22222222	22222222	22222222	22222222
33333333	33333333	33333333	33333333	33333333	33333333	33333333	33333333
44444444	44444444	44444444	44444444	44444444	44444444	44444444	44444444
55555555	55555555	55555555	55555555	55555555	55555555	55555555	55555555
66666666	66666666	66666666	66666666	66666666	66666666	66666666	66666666
77777777	77777777	77777777	77777777	77777777	77777777	77777777	77777777
88888888	88888888	88888888	88888888	88888888	88888888	88888888	88888888
99999999	99999999	99999999	99999999	99999999	99999999	99999999	99999999
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
11111111	11111111	11111111	11111111	11111111	11111111	11111111	11111111
22222222	22222222	22222222	22222222	22222222	22222222	22222222	22222222
33333333	33333333	33333333	33333333	33333333	33333333	33333333	33333333
44444444	44444444	44444444	44444444	44444444	44444444	44444444	44444444
55555555	55555555	55555555	55555555	55555555	55555555	55555555	55555555
66666666	66666666	66666666	66666666	66666666	66666666	66666666	66666666
77777777	77777777	77777777	77777777	77777777	77777777	77777777	77777777
88888888	88888888	88888888	88888888	88888888	88888888	88888888	88888888
99999999	99999999	99999999	99999999	99999999	99999999	99999999	99999999
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000



Are your files in a  
mess? Get them  
sorted out with this  
program by Les Allen.



# D·I·S·K ORDERLY

WHEN SAVING FILES TO DISK, it is nice to retain an orderly sequence so as to keep a group of files together. However, when a file is extracted from the disk a "gap" is left in the directory table that the next save is made to that location. This means that a subset of associated files are being made the sequence is jumbled.

Let's start on the market with an alpha character that enables a key routine to be applied to the directory but again the sequence can be lost as the routine used is an alpha-numeric sort and not a manual one.

With this utility the order can be decided upon by the user simply by swapping one two local ones at a time. Unlike several sequence is obtained (D4) it may take several operations to get the directory sequence but it works and works well.

The utility works by reading the directory contents into several arrays to store the name, track and sector, file type and file date which is then displayed on the screen for editing. Duplicates of the same name are allowed in the number of entries allowed is 40. Simply select the two locations to be swapped, repeating the process for as long as necessary, and select "or" to write the new directory.

## Functional Listing

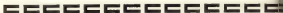
- 10 - 15 clear screen and set up colours
- 16 - 25 variables for screen table
- 26 - 35 print screen and look for RETURN key pressed
- 36 - 45 sets array variables for track/sector/buffer pointer
- 46 - 55 initialise drive
- 56 - 65 opens a new disk file
- 66 - 75 get name from track 10/sector 0 and print to screen
- 76 - 85 dimensions arrays
- 86 - 100 read each directory entry and check for an open file
- 110 - 120 print each name to the screen
- 140 - 150 manual sort routine
- 160 - 180 write new directory to disk
- 180 - 200 disk error channel test
- 210 - 230 number of characters to be read from buffer
- 240 - 250 read character from buffer
- 300 - 305 get OK to continue

## Program Listing

```

10 REM      DIRECTORY ORDERLY
11
12
13 FORS3200,15:PORES3201,6
14 PRINTCHR$(S3200)(147)
15 T1=CHR$(117):FORX=1TO20:T1#T1
16 #CHR$(96):NEXT T1:T1#T1+CHR$(106)
17 T2=CHR$(125):FORX=1TO20:T2#T2
18 #CHR$(32):NEXT T2:T2#T2+CHR$(125)
19 T3=CHR$(125)+" DIRECTORY ORDERLY "+CHR$(125)
20 T4=CHR$(125)+" BY LES ALLEN "+CHR$(125)
21 T5=CHR$(106):FORX=1TO20:T5#T5
22 #CHR$(96):NEXT T5:T5#T5+CHR$(107)
23 PRINT:PRINTSPC(118):PRINTSPC(8)
24 T6=PRINTSPC(8)T5
25 PRINTSPC(8)T6:PRINTSPC(8)T6:P
26 INTSPC(8)T6:PRINTSPC(8)T6

```





## Program Listing - (cont.)

```

80 PRINT#15,"B-P: ",2,BP
81 DO I=1-A+23:GOSUB270:JOS=25
82 PRINTCHR$(15);FORJ=1TOJ1:PRINT:
NEXT
83 PRINTSPC(11);"DISK NAME 10 DON"
84 PRINT:PRINTSPC(10)
90 PRINT:PRINT:PRINT
92 :
94 REM ***** GET UP D
IRECTORY ALWAYS *****
96 :
97 DIR$(15);PTR(152);T$(152);N$(
152);F$(152);S$(152);T$(15
80 T$(1)="$B$";T$(2)="$P$";T$(3)="$
J$";T$(4)="$M$
92 FORI=1TO30:JOS=JOS+CHR$(0);NEXT
94 :
96 REM ***** READ DIR
ECTORY ENTRIES *****
98 :
99 S=1;E=1
100 PRINT#15,"U1: ",2,D;T;S:GOSUB280.
BP=1
104 PRINT#15,"B-P: ",2,BP:GOSUB280;S
A(5)=A: REM GET NEXT SECTO
R
106 PRINT#15,"B-P: ",2,BP:GOSUB280;F
T=A: REM GET FILE TYPE
108 IFPT=DIRPT=128THENBP=BP+31:GOTO
122
109 IFPT=128THENPRINTSPC(8);CHR$(11
);"FILE ERROR NOT CLOSED":GOTO252
109 FT$(E)=FB
104 A=P:GOSUB270:T$(E)=25:
REM GET TAG
106 A=15:GOSUB270:N$(E)=25:
REM SET NAME
108 A=11:GOSUB270:F$(E)=25:
REM SET FILE
110 :
112 REM ***** PRINT
IRECTORY ENTRY *****
114 :
116 PRINTSPC(30);F$(E) " " ;T$(E);1
250
118 PRINTCHR$(145);CHR$(145)
120 BP=BP+2;E=E+1
122 IFBP=255GOTO98
124 IFB$(E) <> 255THENB$(E)=S(5):GOTO98
126 FORJ=1TO1000:NEXT:GOSUB300
128 E=E+1:IFE=68GOTO204
130 :
132 REM ***** HANJIA
L SORT ROUTINE *****
134 :
136 IFB(40)HENDGOTO134
138 PRINTCHR$(147)
140 FORI=1TOINT((E/2)+.5):PRINTWRIGHT
$(STR$(I);2) " "N$(I):NEXT
142 PRINTCHR$(18)
144 FORI=INT((E/2)+.5)+1TOE:PRINT..R
$(STR$(I);2) " "N$(I):NEXT
146 PRINTCHR$(18);FORJ=1TO21:PRINT
NEXT
148 PRINTSPC(3);"SWAP OVER WHICH NU
MBERS * TO END"
150 PRINTSPC(11);"FROM .. TO .
.."
152 GETKEYS:IF(KEY$="O"ORKEY$="S")
ANDKEY$ <> ""THEN157
154 IFKEY$=""THENR04
156 PRINTCHR$(145);SPC(16);KEY$;A=U+
L(KEY$)
158 GETKEYS:IF(KEY$="O"ORKEY$="S")
ANDKEY$ <> CHR$(13)THEN158
160 IFKEY$ <> CHR$(13)THENPRINTCHR$(
145);SPC(17);KEY$;A=A+10=VAL(KEY$)
162 IFA=109A=ETHEX16
164 GETKEYS:IF(KEY$="O"ORKEY$="S")
ANDKEY$ <> ""THEN154
166 IFKEY$=""THENR04
168 PRINTCHR$(145);SPC(27);KEY$;B=U+
L(KEY$)
170 GETKEYS:IF(KEY$="O"ORKEY$="S")
ANDKEY$ <> CHR$(13)THEN170
172 IFKEY$ <> CHR$(13)THENPRINTCHR$(
145);SPC(28);KEY$;B=B+10=VAL(KEY$)
174 IFB=109B=ETHEX16
176 A$=PTR(A);PTR(A)=PTR(B);PTR(B)
=A$
178 A$=T$(A);T$(A)=T$(B);T$(B)
=A$
180 A$=N$(A);N$(A)=N$(B);N$(B)
=A$
182 A$=F$(A);F$(A)=F$(B);F$(B)
=A$
184 A$=S$(A);S$(A)=S$(B);S$(B)
=A$
186 PRINTCHR$(145);SPC(31);"ANOTHER
ENTRY Y/N?";
188 GETKEYS:IFKEY$ <> "Y"ANDKEY$ <> "N"
THEN158

```





## Program Listing - (cont.)

```

190 IFKEY$="Y"THEN130
192 GOTO204
194 PRINTCHR$(146)*SPC(7)*"DO MANY
FILES FOR MANUAL",FORX=1TO2000:NEXT
X
196 GOTO252
198 -
200 REM ***** WRITE 0
IRECTORY ENTRIES *****
202 -
204 PRINTCHR$(147)*"WRITING ... : "
108:PRINT
208 Z$=CHR$(0)+CHR$(255)
208 S=1:N=1
210 PRINT#15,"L1:"Z;D;T;S;DOSL252
212 BP=2
214 IFN=GOTO230
216 IFBP=255GOTO232
218 PRINT#15,"L2:"Z;D;T;S;DOSL252
220 S=S+1:GOTO210
222 PRINT#15,"B-P:";Z;BP
224 PRINTSPC(14)*PRCN)" "THASDC
FISCN)+100)
226 PRINT#2,FISCN);ISSCN);MFCN);P
BCHN);
228 BP=BP+32:N=N+1:GOTO214
230 IFBP=255GOTO238
232 PRINT#15,"B-P:";Z;BP
234 PRINT#2,Z$
236 BP=BP+32:GOTO230
238 BP=0:PRINT#15,"B-P:";Z;BP
240 PRINT#2,Z$
242 PRINT#15,"L3:"Z;D;T;S;DOSL252
244 Z$=LEFT$(Z$,2)
246 S=S+1:GOTO210
248 PRINT#15,"L1:";Z;D;T;S;DOSL252
Z
250 BP=2:GOTO232
252 FORX=1TO1000:NEXT:CLOSE2,CLOSE
15
254 PRINTCHR$(147)*"PORT-1TO11:PRIN
T:NEXT:PRINTSPC(11)*"ANOTHER DISK
Y/N?"
256 GETKEY$:IFKEY$>"Y"ASKEY$>"N
"THEN258
258 IFKEY$="Y"THEN130
260 POKE3250,14:PRINTCHR$(147)*CHR
$(154):END
262 -
264 REM ***** INPUT 5F
ROR CHANNEL STATUS *****
266 -
268 INPUT#15,EN,DMS,ET,ES:IFEN=0TH
ENRETURN
270 PRINTCHR$(147)*"DISK ERROR "EN,
DMS,ET,ES;GOTO252
272 -
274 REM ***** SET CHAR
ACTERS FROM BLF *****
276 -
278 Z$=""
280 FORI=1TO4:DOSL250
282 Z$=Z$+B;NEXT:RETURN
284 -
286 REM ***** INPUT
I FROM BUFFER *****
288 -
290 GET#2,AS:IFAS=" "THEN#3+CHR$(0)
292 A=ASC(AS):BP=BP+1:RETURN
294 -
296 REM ***** GET D
K TO CONTINUE *****
298 -
300 PRINTSPC(10)*CHR$(11)*"OK TO CONT
INUE Y/N?"
302 GETKEY$:IFB$="N"GOTO252
304 IFKEY$>"Y"GOTO306
306 RETURN
308 -
310 -
312 *****
314 *
316 * DISK DIRECTORY ORD
ONLY FOR COMMODE 64
317 *
318 *
319 * SAMP ANY 2 DIRECTO
RY ENTRIES AT A TIME
320 *
321 * MAXIMUM NUMBER OF
ENTRIES ALLOWED := 14
322 *
323 *
324 * LIES ALLAN
19.01.85
325 *
326 *
327 *****

```





**J. FITZPATRICK (MC),**  
1 Albany Ave., Waltham,  
Massachusetts 02154



1. *Staphylococcus aureus* (Gram +)  
 2. *Staphylococcus epidermidis* (Gram +)  
 3. *Staphylococcus saprophyticus* (Gram +)  
 4. *Staphylococcus carnosus* (Gram +)  
 5. *Staphylococcus sciuri* (Gram +)  
 6. *Staphylococcus hyicus* (Gram +)  
 7. *Staphylococcus pasteuri* (Gram +)  
 8. *Staphylococcus saprophilus* (Gram +)  
 9. *Staphylococcus lentus* (Gram +)  
 10. *Staphylococcus saprophilus* (Gram +)

Robert R. Kohn, Jr., Editor  
Journal of Management Education  
Sage Publications, Inc.  
Los Angeles, CA 90024

For more information, see page 100-10. **STRENGTHENED FRAME** Type concrete and steel reinforcement is used in the frame of wall units, resulting in better resistance to impact and vibration. This feature, combined with specially designed insulation, including polyurethane insulation, allows greater structural loads and soundproofing, the optimum strategy for recording from other programs. The soundproof lining is made in the sound chamber while the unit is being assembled in the factory. When it is being used for recording, the chamber can be completely sealed and insulated. Sound transmission is reduced, and sound absorption is increased, and

© 1999 by The McGraw-Hill Companies, Inc.

THE UNIVERSITY OF CHICAGO PRESS

**DON'T BE AN  
ALIEN  
TO INVADE THIS  
SPACE  
RING  
01 437 0699**

[illegible]

**YOUR COMMODORE — CLASSIFIED ADVERTISEMENT — ORDER FORM**

[illegible]

ADVERTISERS PLEASE ADD 15% VAT

Advertise nationally in these columns for only 40p per word (min charge 16 words). Simply print your message in the coupon and send with your cheque or postal order made payable to Argus Specialist Publications Ltd to:

**CLASSIFIED DEPARTMENT**  
**YOUR COMMODORE**  
No 1 Golden Square, London W1  
Tel: 01-437 0880

\_\_\_\_\_

1999, 2000, 2001, 2002, 2003, 2004, 2005, 2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016, 2017, 2018, 2019, 2020, 2021, 2022, 2023, 2024, 2025, 2026, 2027, 2028, 2029, 2030, 2031, 2032, 2033, 2034, 2035, 2036, 2037, 2038, 2039, 2040, 2041, 2042, 2043, 2044, 2045, 2046, 2047, 2048, 2049, 2050, 2051, 2052, 2053, 2054, 2055, 2056, 2057, 2058, 2059, 2060, 2061, 2062, 2063, 2064, 2065, 2066, 2067, 2068, 2069, 2070, 2071, 2072, 2073, 2074, 2075, 2076, 2077, 2078, 2079, 2080, 2081, 2082, 2083, 2084, 2085, 2086, 2087, 2088, 2089, 2090, 2091, 2092, 2093, 2094, 2095, 2096, 2097, 2098, 2099, 2100, 2101, 2102, 2103, 2104, 2105, 2106, 2107, 2108, 2109, 2110, 2111, 2112, 2113, 2114, 2115, 2116, 2117, 2118, 2119, 2120, 2121, 2122, 2123, 2124, 2125, 2126, 2127, 2128, 2129, 2130, 2131, 2132, 2133, 2134, 2135, 2136, 2137, 2138, 2139, 2140, 2141, 2142, 2143, 2144, 2145, 2146, 2147, 2148, 2149, 2150, 2151, 2152, 2153, 2154, 2155, 2156, 2157, 2158, 2159, 2160, 2161, 2162, 2163, 2164, 2165, 2166, 2167, 2168, 2169, 2170, 2171, 2172, 2173, 2174, 2175, 2176, 2177, 2178, 2179, 2180, 2181, 2182, 2183, 2184, 2185, 2186, 2187, 2188, 2189, 2190, 2191, 2192, 2193, 2194, 2195, 2196, 2197, 2198, 2199, 2200, 2201, 2202, 2203, 2204, 2205, 2206, 2207, 2208, 2209, 2210, 2211, 2212, 2213, 2214, 2215, 2216, 2217, 2218, 2219, 2220, 2221, 2222, 2223, 2224, 2225, 2226, 2227, 2228, 2229, 2230, 2231, 2232, 2233, 2234, 2235, 2236, 2237, 2238, 2239, 2240, 2241, 2242, 2243, 2244, 2245, 2246, 2247, 2248, 2249, 2250, 2251, 2252, 2253, 2254, 2255, 2256, 2257, 2258, 2259, 2260, 2261, 2262, 2263, 2264, 2265, 2266, 2267, 2268, 2269, 2270, 2271, 2272, 2273, 2274, 2275, 2276, 2277, 2278, 2279, 2280, 2281, 2282, 2283, 2284, 2285, 2286, 2287, 2288, 2289, 2290, 2291, 2292, 2293, 2294, 2295, 2296, 2297, 2298, 2299, 2300, 2301, 2302, 2303, 2304, 2305, 2306, 2307, 2308, 2309, 2310, 2311, 2312, 2313, 2314, 2315, 2316, 2317, 2318, 2319, 2320, 2321, 2322, 2323, 2324, 2325, 2326, 2327, 2328, 2329, 2330, 2331, 2332, 2333, 2334, 2335, 2336, 2337, 2338, 2339, 2340, 2341, 2342, 2343, 2344, 2345, 2346, 2347, 2348, 2349, 2350, 2351, 2352, 2353, 2354, 2355, 2356, 2357, 2358, 2359, 2360, 2361, 2362, 2363, 2364, 2365, 2366, 2367, 2368, 2369, 2370, 2371, 2372, 2373, 2374, 2375, 2376, 2377, 2378, 2379, 2380, 2381, 2382, 2383, 2384, 2385, 2386, 2387, 2388, 2389, 2390, 2391, 2392, 2393, 2394, 2395, 2396, 2397, 2398, 2399, 2400, 2401, 2402, 2403, 2404, 2405, 2406, 2407, 2408, 2409, 2410, 2411, 2412, 2413, 2414, 2415, 2416, 2417, 2418, 2419, 2420, 2421, 2422, 2423, 2424, 2425, 2426, 2427, 2428, 2429, 2430, 2431, 2432, 2433, 2434, 2435, 2436, 2437, 2438, 2439, 2440, 2441, 2442, 2443, 2444, 2445, 2446, 2447, 2448, 2449, 2450, 2451, 2452, 2453, 2454, 2455, 2456, 2457, 2458, 2459, 2460, 2461, 2462, 2463, 2464, 2465, 2466, 2467, 2468, 2469, 2470, 2471, 2472, 2473, 2474, 2475, 2476, 2477, 2478, 2479, 2480, 2481, 2482, 2483, 2484, 2485, 2486, 2487, 2488, 2489, 2490, 2491, 2492, 2493, 2494, 2495, 2496, 2497, 2498, 2499, 2500, 2501, 2502, 2503, 2504, 2505, 2506, 2507, 2508, 2509, 2510, 2511, 2512, 2513, 2514, 2515, 2516, 2517, 2518, 2519, 2520, 2521, 2522, 2523, 2524, 2525, 2526, 2527, 2528, 2529, 2530, 2531, 2532, 2533, 2534, 2535, 2536, 2537, 2538, 2539, 2540, 2541, 2542, 2543, 2544, 2545, 2546, 2547, 2548, 2549, 2550, 2551, 2552, 2553, 2554, 2555, 2556, 2557, 2558, 2559, 2560, 2561, 2562, 2563, 2564, 2565, 2566, 2567, 2568, 2569, 2570, 2571, 2572, 2573, 2574, 2575, 2576, 2577, 2578, 2579, 2580, 2581, 2582, 2583, 2584, 2585, 2586, 2587, 2588, 2589, 2590, 2591, 2592, 2593, 2594, 2595, 2596, 2597, 2598, 2599, 2600, 2601, 2602, 2603, 2604, 2605, 2606, 2607, 2608, 2609, 2610, 2611, 2612, 2613, 2614, 2615, 2616, 2617, 2618, 2619, 2620, 2621, 2622, 2623, 2624, 2625, 2626, 2627, 2628, 2629, 2630, 2631, 2632, 2633, 2634, 2635, 2636, 2637, 2638, 2639, 2640, 2641, 2642, 2643, 2644, 2645, 2646, 2647, 2648, 2649, 2650, 2651, 2652, 2653, 2654, 2655, 2656, 2657, 2658, 2659, 2660, 2661, 2662, 2663, 2664, 2665, 2666, 2667, 2668, 2669, 2670, 2671, 2672, 2673, 2674, 2675, 2676, 2677, 2678, 2679, 2680, 26

**Tel no idnek...**

Please place my ad in **YOUR COMPANY** by \_\_\_\_\_

© 2000 Blackwell Science Ltd, *Journal of Internal Medicine* 247: 105–112







# WIZARD COMPUTER GAMES

A NEW EXCITING ADVENTURE GAME By Tony Crowther

£1000 Treasure to be won & a free disk with every game

Available for Commodore 64, Spectrum 48 (from October 1991) (also available for Amstrad)

## WILLIAM WOBBLER



William Wobbler is the latest game from Tony Crowther—a most exciting adventure game. Through the underworld of dark powers in search of golden treasure William struggles against all odds to vanquish foes and reach his goal. A game of skill and excitement.





# Choosing the right computer is a good start — but can you find the right software?



At SUPERSOFT we're very conscious of the fact that people who spend several hundred pounds on computer equipment are looking to do rather more than play space invaders.

Financial planning is a rather grand name for something you're been doing all your life — making ends meet! Perhaps if Mr McWorster had used **BUSICALC** he would have been able to balance the books a little better.

For home, club or small business use **BUSICALC 1** should pay for itself in no time at all; for larger companies we recommend **BUSICALC 2**, one of the few really valuable programs that you can learn to use in a day.

Although your Commodore 64 is a power, full musical instrument you need to tie a pretty good programmer to understand how it all works. Unless, of course, you buy **MUSIC MASTER**.

To use **MUSIC MASTER** requires no prior musical knowledge, though in the hands of an experienced musician it will prove an invaluable tool. You don't need to know the first thing about programming when **MUSIC MASTER** is the musical equivalent of a word processor, remembering the notes you play and allowing you to replay and edit them as you wish.

**INTERSECTION PILOT** is a space flight simulator. Nowadays simulators are widely used to train pilots and astronauts because — to be frank — it's a lot cheaper (and safer) than the real thing!

Imagine, if you will, life in the 23rd century: space travel is commonplace, and on the outskirts of the galaxy the first war between civilisations is being fought. A heritage of trained pilots has prompted the Federation to develop a computer simulation that allows new recruits to gain experience without paying for their mistakes with their lives. With the aid of your Commodore 64 you too can learn to pilot the *Interceptor Mk 3* craft. But be warned — this is no game!

Other SUPERSOFT products include the **MICRO ASSEMBLER** cartridge, the only assembler that's ideal for beginners yet powerful enough for the professional (most of our computers use it); the **VICTORY** cartridge, with dozens of tournaments to back including tactics, sets, analysis commands, or on disk there's **MASTERS 64**, a really comprehensive package for the keen ping-pong player.

Of course, we do also publish games programs, and with classics like **STUD**, **QUIPO** and **RAMBLER** in our range we are one of the market leaders. But we must stop coming up with the list of programs that are going to be in use for months and years, not hours and days — the sort of programs that make you glad that you bought a computer — and glad that you bought SUPERSOFT!

You won't find SUPERSOFT products on the shelves of your local supermarket. But most specialist shops stock titles from our extensive range (and are prepared to obtain other programs to order). However you can also buy direct by sending a cheque/pre-paid order or just *telex*, by calling at our offices, or over the telephone using your ACCESS card.



SUPERISOFT, Winchester House, Canning Road,  
Walsingham, Norfolk, NR25 6AA  
Telephone: 01-261 1166